Introduction to LaTeX Workshop

In aid of Our Lady's Children Hospital Crumlin, Research Foundation (C.M.R.F). The purpose of this workshop is to give students the basic tools in order to create documents that are required throughout the Maths, ACM and Physics pathways. This workshop will provide tools to create a platform in which can be further expanded on by the individual in LaTeX, but will provide the skills to create Lab Reports, CV's and Notes.

Created By

Owen Johnson
Ben Graham
Hugh Lucey-Kelly

In association with
University College Dublin
Physics & Maths Department

CMRF | CRUMLIN
Every sick child deserves every chance

UCD SCIENCE SOCIETY

2019

Acknowledgements

We would like to thank Dr. Sharon Shannon and Prof. Peter Duffy of the Physics department for the guidance garnered throughout the creation of this project whether it was direct advice in regards to the project or the teaching of skills that allowed such a project to come to full fruition.
We would also like to acknowledge our peers in the 2018 - 2022 MPG classes for harbouring a co-operative environment and offering much needed advice when approaching how this workshop should be delivered.
and as always the staff and members in the C.M.R.F who carry out paramount work, truly giving all of Ireland's sick children every chance.

Sincerely, Owen & the 2019/20 Science Society Committee.

Contents

1 What is LaTeX?

Brought to fruition in 1984, LaTeX is a typesetting program which makes use of the TeX typesetting language. It has proven extraordinarily popular amongst Physicists, Mathematicians and Computer Scientists with many users from other backgrounds in academia. Its features can streamline and automate a lot of tedious tasks with the use of packages and command lines. It can also produce concise arrays of mathematical equations, tables and figures, for example, while maintaining the desired format of the user, no matter the size or complexity of the project.

2 Compiling LaTeX code

2.1 Compilers

There are many different ways of compiling LaTeX code. A commonly used compiler amongst beginners and intermediate users is a web based compiler called 'Overleaf'. Overleaf is completely cloud based so it will run in any browser environment. It is free to use with some quality of life features locked behind a premium fee.
For the purposes of this workshop and its simple and fast setup Overleaf will be used in examples, with a few exceptions.

Alternatives to Overleaf;

- TeX Studio
  - Free
  - Open-Source
  - Unix-based OS, MAC OS and Windows platforms
  - Powerful
  - Highly customisable
  - Difficult learning curve
  - Clunky
- TeX Creator
  - Free
  - Open-Source
  - Unix-based and Windows platforms
- TeX Works
- MikTeX

2.2 File Types

Throughout this workshop we will only be dealing with two common file types used within Latex. .tex files and .bib files.
.tex files are used as the source files for Latex. They are used to initiate document outputs and run various graphic or formatting commands.
.bib files are used as directories for referencing tools such as biblatex and bibtex.

2.3 Packages in LaTeX

Much like other popular programming languages, such as Python, LaTeX uses developer and community-made packages to define functions and automate various useful tools.
To import a package, the following string is placed before the command to begin document, `\usepackage{packagename}`
Most common packages used when writing maths-heavy documents are packages such as amsmath, siunitx, graphicx, wrapfig, listings and multicol

Listing 1: Example of some of the packages used in a **PHYC20020** lab report

```
\documentclass[12pt]{article}
\usepackage[a4paper]{geometry}
\usepackage[myheadings]{fullpage}
\usepackage{fancyhdr}
\usepackage{lastpage}
\usepackage{graphicx, wrapfig, subcaption, setspace, booktabs}
\usepackage[T1]{fontenc}
\usepackage[font=small, labelfont=bf]{caption}
\usepackage{fourier}
\usepackage{url, lipsum}
\usepackage{amsmath}
\usepackage{subfiles}
\usepackage{gensymb}
\usepackage{siunitx}
\usepackage{url}
\usepackage{multicol}
\usepackage{amsmath}
\usepackage{wrapfig}
\usepackage{listings}
\usepackage{caption}
```

.

3 Document Structure

3.1 Creating documents & defining parameters

3.1.1 Types of documents

Three of the most common document classes in Latex are *book, article* and *report*. These document types will define arbitrary things about the document such as page size, font size, columns and so on. The document class is the first thing that gets defined using the following command, `\documentclass{}`. These document classes' attributes can be individually modified by specifying certain parameters as seen in the following example.

```
\documentclass[twoside, openright, a4, 12pt]{report}
\begin{document}
\input{main.tex}
\end{document}
```

Attributes here are numerous and the specificity of all possible document class modifiers can be found on the official 'Project Latex' website. Examples of what parameters placed on the document class can also be found in the sample code found in the source code.

3.2 Geometry of Document

3.2.1 Margins

Within the preamble in LaTeX using the **geometry** package will allow modification to the boundaries of the text using the following command, `\newgeometry{left= cm, right= cm, top= cm, bottom= cm}`. This command will change the margins on the specific page it's executed on and any subsequent pages after unless it is redefined. The margins can be in metric units, or also determined as a factor of certain aspects of the document such as `\linewidth` and `\textwidth`. It is also common to use the **fullpage** package instead of the geometry package. This can be particularly useful for lab reports as the page size it provides fits the lab report style.

- ⋆ Whenever we talk about a distance there are a few different units that LaTeX recognises. The most common are: mm, cm, pt and ex (one ex is the height a line of text). ⋆

3.2.2 Centering

The most common ways to center text within LaTeX, are using the `\centerline{text}` and `\centering` commands. For larger sections of text that need to be centered the centering environment is will produce and be initiated by the following.

Inputted text will be centered as such.

```
\begin{center}
    \textit{Inputted text }
\end{center}
```

3.2.3 Spacing

Sometimes it's necessary to create regions of space within a document, this can be done vertically and horizontally by `\vspace{ distance }` and `\hspace{ distance }`, where the inputted parameter is the desired distance. `\vfill` is also a useful command that will fill the rest of the current page with a white space. Any text placed after `\vfill` will be placed at the bottom of the specified page.

Breaking pages is quite simple in LaTeX, it is done using the `\newpage` command. Note that LaTeX automatically arranges text into pages, taking care of possible overflow.

3.3 Title Page Creation

Title pages are denoted at the start of the document either the `\maketitle` command or the title page environment as seen in the below listings. In LaTeX the user has a lot of control over the geometry and layout of a title page. It's something that a sufficient amount of time can be spent tweaking. For further examples of this reference the sample code and see the further reading section of the document.

Listing 2: Sample of a title page.

```
\begin{titlepage}
   \begin{center}
       \vspace*{1cm}

       \textbf{reportTitle}

       \vspace{0.5cm}
        reportSubtitle

       \vspace{1.5cm}

       \textbf{Author Name}

       \vfill

       A reportpresented for the degree of\\
       Doctor of Philosophy

       \vspace{0.8cm}

       \includegraphics[width=0.4\textwidth]{university}

       Department Name\\
       University Name\\
       Country\\
       Date

   \end{center}
\end{titlepage}
```

3.4 Section creation

Like most documents its required to split things into categories. Depending on the document class in LaTeX the sections are prioritized differently the general hierarchy is as follows,

1. Chapters
2. Sections
3. Sub-Sections
4. Subsub-Sections

Each inputted respectively as follows, `\chapters{name}`, `\sections{name}`, `\subsection{name}` and `\subsection{name}`. To remove the number from the title and table of contents an asterisks is placed just after the initial command as follows, `\command*{}`.

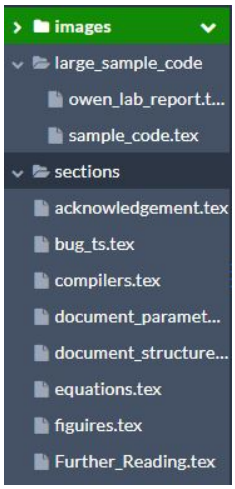
3.5 Table of Contents

LaTeX does a really good job of managing how the table of contents is handled. It's very simple to place within the document, using the following command `\tableofcontents`. How detailed the table of contents is can be influenced

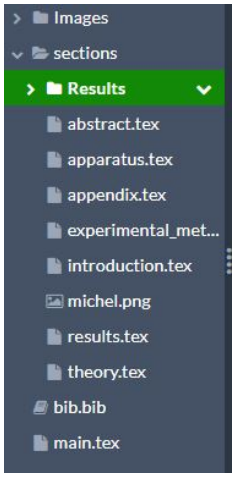
by the following command, `\setcounter{tocdepth}{2}`, with the number in the curly bracket being the depth (Ie. are only sections included? Sections and subsections?).
It's possible to mess around with the styles and layout of the table of contents. It's outside of the scope of content for this workshop but guides on how this is done can be found in the further reading section.

3.6 Organizing project's using different files

It's highly useful to spread work across many different .tex files when working on larger projects such as notes for a class or a lab report. For example the two images below are what the project files looked like for the notes for this workshop and a PHYC20020 Lab Report.



(a) Workshop Notes



(b) Lab Report

Figure 1: Overview of files used in two common projects

To do this create and title the desired file type, such as a .tex file, then place it in the directory of where the projects main tex file is being compiled

3.7 Placing comments within a project

Like other programming languages its important to have a clear structure when navigating code. To place a comment within your file the % is used. Anything placed after % will be diregarded as a comment until the next sign is started.

4 Inputting Text

4.1 Basic Text Input

Text input in LaTeXis simple, it's just a case of inputting the desired text as seen in the following example,

Listing 3: Example of a passage of text.

```
\section{2011 Revisions}
The 2011 ZX-10R underwent major overhaul both mechanically and visually. Most
   notably, Kawasaki introduced their Sport Kawasaki Traction Control (S-KTRC)
   system as standard. \\
It predicts when traction will be lost and adapts accordingly. Also new are an
   ABS option called Kawasaki Intelligent Braking System (KIBS), a completely
   new design, adjustable foot-pegs, larger throttle bodies, a horizontal rear
   suspension, lighter three-spoke wheels, Showa Big Piston Fork (BPF)[3] front
    suspension, and an LCD panel dashboard.[4] \\
The 2012 model is identical to the 2011 with the only exception being the
   slightly different paint schemes offered. In 2013 the models went under
   another small revision where the colors offered changed again and the front
   damper was replaced with an Ohlins electronic front steering dampener. In
   2014 the only changes were different colors. In 2015 the only color offered
   was black, alongside a 30th anniversary edition (in EU, a green color is
   also offered).
```

As well as simply imputing the text, if you want to move onto a new line two backslashes \\ are placed at the end of the last line as shown in the example above. If you type two backslashes and then leave a blank line in the code you will start a new paragraph which will be indented. A 'double enter' will leave a similar spacing but not indent the paragraph.

4.2 Bold, Italics, Underlining and Others

Much like in word it's possible to underline, **embolden**, *italicise* text. It's quite simple to do within LaTeX. It can be completed with the following commands.

Command	Output
`\textbf{}`	**bold**
`\textit{}`	*Italic*
`\underline{}`	Underline
`\texttt{}`	`type text`
`$\overline{}$`	$\overline{Overline}$

⋆ It is important to notice that the 'overline' command can only be used within a maths environment.⋆

It is important to familiarize yourself with the hotkeys associated with your compiler and define hotkeys for functions that are used regularly as it will drastically improve the time it takes to produce sections within a document or project.
There are also many more styles the fonts can be given within the text that will be included on the condensed note sheet.

4.3 Lists

Out of the box LaTeX supports two types of lists, *enumerated* lists and *itemized* lists. The latter is the standard bullet point variant of a list with the former being a numbered list. To begin a list `enumerate` and `itemize` are used to initialize the desired list environment.

An example of a simple numerated list is as follows.

1. Item
2. Item
3. Item
4. Item

Listing 4: Example of a enumerated list

```
\begin{enumerate}
    \item Item
    \item Item
    \item Item
    \item Item
\end{enumerate}
```

To produce the following nested list the code in the below listing is used,

1. Item
    (a) Item
    (b) Item
    (c) Item
2. Item
    (a) Item
    (b) Item
    (c) Item
3. Item
    (a) Item
    (b) Item
    (c) Item
4. Item
    (a) Item
    (b) Item
    (c) Item

Listing 5: Example of a nested list.

```
 \begin{multicols}{2}
 \begin{enumerate}
     \item Item
     \begin{enumerate}
         \item Item
         \item Item
         \item Item
     \end{enumerate}
     \item Item
     \begin{enumerate}
         \item Item
         \item Item
         \item Item
     \end{enumerate}
\end{multicols}
```

As seen above it's possible to nest lists within lists to create any amount of sub-levels desired.

4.4 Fonts & Aesthetics

4.4.1 Colors

To color text within LaTeX the **color** package is required. Once package is executed in the preamble any color can then be used to in text like so. This is done using the `{\color{colour_name}text}` command. If the color package

does not contain the desired color needed in your document it may be in the **xcolor** package which has an extensive range of colors.[1]. Its also possible to define the color code using color codes from the RGB range using the following command in the preamble `\definecolor{neonGreen}{rgb}{17,255,0}` .

4.4.2 Fonts

LaTeX by default uses an elegand sans serif subsidary of the computer modern font family. However, the fonts used within LaTeX are easily changed.

To change the font of an entire document a command is added to the preamble just like a package, many user created and source fonts can be found on the LaTeXfont catalog. `https://tug.org/FontCatalogue/`.

To change the font of a specific section of text like so,

Softly, in the dusk, a woman is singing to me;
Taking me back down the vista of years, till I see A child sitting under the piano,
in the boom of the tingling strings And pressing the small,
poised feet of a mother who smiles as she sings.

The above passage is obtained by the following input,

```
\begin{center}
{\fontfamily{qzc}\selectfont

Input of desired text.

}
\end{center}
```

4.4.3 Font sizes

It's easy to change the size of fonts of specific passages of text relative to the size of the base font defined in the preamble. Much like the color command, `{\size input text here}`. Sizes and their respective output can be found below.

`\tiny` produces tiny

`\scriptsize` produces scriptsize

`\footnotesize` produces footnotesize

`\small` produces small

`\normalsize` produces normal

`\large` produces large

`\Large` produces Large

`\LARGE` produces LARGE

`\huge` produces huge

`\Huge` produces Huge

[1]See further reading for more information

5 Tables

5.1 Inserting Tables

The `tabular` command is used to typeset tables within LaTeX, it's done by creating an environment as follows with `\begin{tabular}{---}`. The parameter following the initial command is used to define the format of the columns.

- l will place any entries in a specific column to the left
- r will place any entries in a specific column to the right
- c will place any entries in a specific column to the center
- with | placing a line specified with which side of a specific column its on

To insert lines between the different rows in a table the command `\hline` is used. It can also be used twice to add a double line and so on.
When creating a table we place the whole tabular environment inside a table environment by using `\begin{table}[h]`. The '[h]' is added after so that the table will be placed in the closest possible place to where it is in the code. (There are alternatives to h - there is a list of these in the figures chapter in section 6.2)

Using all of this we can make the following table,

Table 1: TableName

n	4.75	4.8	4.8	5.05
yeet theorem	n/a	n/a	n/a	n/a
average	2	3	4	5
decay	3.692	4.683	5.683	7.733

Which is given by the following block of commands,

```
\begin{table}[h]
\centering
\caption{TableName}
\begin{tabular}{|l|l|l|l|l|}
\hline

n & 4.75 & 4.8 & 4.8 & 5.05 \\ \hline
yeet theorem & n/a & n/a & n/a & n/a \\ \hline
average & 2 & 3 & 4 & 5 \\ \hline
decay & 3.692 & 4.683 & 5.683 & 7.733 \\ \hline

\end{tabular}
\end{table}
```

We also used the command `\caption{}` to give the table a title. In a lab report the title for a table is always placed above the table but the caption can be place below the table by placing `\caption{}` just after `\end{tabular}`

A more complex example,

Table 2: Table with aligned units.

Value 1 α	**Value 2** β	**Value 3** γ
1	1110.1	a
2	10.1	b
3	23.113231	c

```
\begin{table}[h!]
  \begin{center}
    \caption{Table with aligned units.}
    \label{tab:table1}
    \begin{tabular}{l|c|r}
      \textbf{Value 1} & \textbf{Value 2} & \textbf{Value 3}\\
      $\alpha$ & $\beta$ & $\gamma$ \\
      \hline \hline
      1 & 1110.1 & a\\
      2 & 10.1 & b\\
      3 & 23.113231 & c\\
    \end{tabular}
  \end{center}
\end{table}
```

Lab report example,
using `\multicolumn{No. of columns to combine}{Format of combined column(s)}{Contents of column(s)}`. `\multicolumn` can be used to combine several adjacent columns. The "Format of combined column(s)" follows the rules of the tabular parameter as described above.

Table 3: Data collected as the total lengths calculated as solved for a convex lens

$f_1\ (cm)$	$f_t\ (cm)$	$\sum f\ (cm)$	$f_2\ (cm)$
0.070 ± 0.004	0.092	0.162	-44.314 ± 0.222
0.071 ± 0.004	0.095	0.166	-41.831 ± 0.209
0.070 ± 0.004	0.094	0.164	-42.533 ± 0.213
0.070 ± 0.004	0.071	0.140	-1052 ± 5.26
0.070 ± 0.004	0.080	0.150	-99.905 ± 0.500
$\bar{f}_2 = -57.146 \pm 0.095\ cm$			
$\sigma_f = 0.474\ cm$			

Produced by following code,

```
\begin{table}[h]
\centering
\caption{Data collected as the total lengths calculated as solved for a convex
    lens}
\begin{tabular}{||c|c|c|c||}
\hline
$f_1 \;(cm)$ & $f_t \; (cm)$ & $\sum f \; (cm)$ & $f_2 \; (cm) $ \\ \hline
0.070 \pm 0.004 & 0.092 & 0.162 & -44.314 \pm 0.222 \\ \hline
0.071 \pm 0.004 & 0.095 & 0.166 & -41.831 \pm 0.209 \\ \hline
0.070 \pm 0.004 & 0.094 & 0.164 & -42.533 \pm 0.213\\ \hline
0.070 \pm 0.004 & 0.071 & 0.140 & -1052 \pm 5.26 \\ \hline
0.070 \pm 0.004 & 0.080 & 0.150 & -99.905 \pm 0.500 \\ \hline
\multicolumn{4}{|c|}{$\bar f_2\; = \; -57.146 \pm 0.095 \; cm$} \\ \hline
\multicolumn{4}{|c|}{\sigma_f = 0.474 \; cm} \\ \hline

\end{tabular}
\end{table}
```

5.2 Importing Tables from CSV Files*

In short the fastest and simplest way to do this while still starting out with LaTeX is using a conversion website such as `https://tableconvert.com/?output=latex`. However there are packages specifically designed to input .CSV files that are in the directory of the main .tex file. This is fast and a great way to implement data changes on a large scale if necessary. However, it can be quite tricky to get working the first couple of uses as the format in

the .CSV file needs to be in line with that of the tabular command within the document.
This can be done using the **csvsimple** package. This is simply done using the `\csvautotabular{filename}` where the comma separated value file is placed in the directory where the file is being constructed or specified otherwise. To import certain columns and modify the .csv import consult the **csvsimple manual** in the further reading.

5.3 Long-tables & Wide tables

Sometimes large tables of data need to be included within a document, an example of this is the Franck-Hertz or Gamma Rays & Counting Stats experiments. To appropriately include this within a document is quite tricky. Two ways to approach this is using a long table or a wide table.

5.3.1 Long-tables

The packages needed for long tables, `\usepackage{longtable}`. This will allow the table to run onto the next page.

Listing 6: Example of a long table

```
\begin{center}
\begin{longtable}{|c|c|}
\caption{Raw Data for Mercury} \label{tab:long} \\

\hline \multicolumn{1}{|c|}{\textbf{$U_b (V)$}} & \multicolumn{1}{c|}{\textbf{$
    I_A (nA)$}}  \\ \hline
\endfirsthead

\multicolumn{2}{c}%
{{\bfseries \tablename\ \thetable{} -- continued from previous page}} \\
\hline \multicolumn{1}{|c|}{\textbf{$U_b (V)$}} & \multicolumn{1}{c|}{\textbf{$
    I_A (nA)$}}  \\ \hline
\endhead

\hline \multicolumn{2}{|r|}{{Continued on next page}} \\ \hline
\endfoot

\hline \hline
\endlastfoot

#DATA INPUT HERE

\end{longtable}
\end{center}
```

5.3.2 Wide Tables

Sometimes a better alternative to long-tables is using wide tables. This is done using the **adjustbox** package. There are two ways which this can be used. One thing this package allows us to is to let the table run into the margins as shown in the following example.

Row No.	Feature	2θ	$\Delta 2\theta$	θ	$\Delta\theta$	$\sin\theta$	$\Delta\sin\theta$	2d	$n\lambda$	$\Delta n\lambda$	n	Δn
1	3	42	0.5	21	0.25	0.3583	0.0836	4.02×10^{-10}	1.44×10^{-10}	3.36×10^{-11}	1	1
2	4	46	0.5	23	0.25	0.3907	0.0899	4.02×10^{-10}	1.57×10^{-10}	3.61×10^{-11}	1	1
3	5	66	0.5	33	0.25	0.5446	0.1141	4.02×10^{-10}	2.19×10^{-10}	4.59×10^{-11}	2	1.4142
4	6	69	0.5	34.5	0.25	0.5664	0.1166	4.02×10^{-10}	2.28×10^{-10}	4.69×10^{-11}	2	1.4142
5	7	94	0.5	47	0.25	0.7313	0.1246	4.02×10^{-10}	2.94×10^{-10}	5.01×10^{-11}	3	1.7320
6	8	98	0.5	49	0.25	0.754	0.1237	4.02×10^{-10}	3.03×10^{-10}	4.98×10^{-11}	3	1.7320

Listing 7: Example code used to create a wide table

```
\begin{table}[h]
\begin{adjustbox}{center}
\begin{tabular}{c c c c c c c c c c c c c}

    *DATA INPUT HERE

\end{tabular}
\end{adjustbox}
\end{table}
```

Alternatively, we can use **adjustbox** to fix the width of the table. **Adjustbox** will make the text smaller instead of going onto a new line, however sometimes this will make the text too small to read.

1	This table will have very small text
2	The more text that is added the smaller the text will be. It will also become more and more difficult to read! The text in this table has been made very small.
3	So this can be useful but only when used carefully.

Listing 8: Example code used to create a wide table

```
\begin{table}[h!]
\adjustbox{max width = \columnwidth}{
\begin{tabular}{|c | c|}
\hline
    *DATA INPUT HERE
\hline
}
\end{tabular}
}
\end{table}
```

5.4 Tables with large amounts of text

Sometimes we need to type tables with large amounts of text in them. However when making tables in LaTeX it automatically makes the table wider until all of the text fits in the cell even if that means that the table runs off the page instead of starting a new line. This problem cannot be fixed by telling LaTeX to go onto a new line as it will think we are creating a new row. There are two solutions to this problem.
The first is to use `p{--}` instead of c,l or r after tabular. This will set the width of that particular column to whatever width we set inside of the brackets. This can be seen in the following example:

Table 4: Table with a lot of text

Text	First Reading	Second Reading
This column can contain a lot of text but no matter how much we type the width of the column will be limited to a maximum of half of the width of a line	1	1
We can continue this on with as many columns as we choose	2	3

This is given by the following commands:

Listing 9: Example of a long table

```
\begin{table}[h]
\begin{center}
\caption{Table with a lot of text}
```

```
\begin{tabular}{|p{0.5\linewidth}|c|c|}
\hline
Text& First Reading & Second Reading\\ \hline
This column can contain a lot of text but no matter how much we type the width
    of the column will be limited to a maximum of half of the width of a line &
    1&1\\ \hline
We can continue this on with as many columns as we choose &2&3\\ \hline
\end{tabular}
\end{center}
\end{table}
```

As we can see in this example the text in any column which doesn't use all of the space is aligned to the top of the cell. To align it to the bottom simply use `b{}` or to align it to the middle use `m{}`.

The other solution to this problem is to use tabularx. To use this we need to add `\usepackage{tabularx}`. The tabularx environment works mostly the same way as the tabular environment with a few small exceptions which are beyond this course. To use the tabularx environment use exactly the same process as with tabular but replace `\begin{tabular}{}` with `\begin{tabularx}{tablewidth}{cellalignment}`. The first bracket is the overall width of the table and the second bracket is for the alignments of the cells as before c,l,r. However when aligning the text in the cells we can also use X. When we use X, LaTeX will make the table as wide as possible within the limitations of the table width we chose, but if there is too much text it will simply move onto a new line. For example:

Table 5: tabularx table

5	some more text is placed here and the space will be divided up equally	here we have some more text which will be spread out
8	We can continue into another column and keep going	This column will also be changed to the best size possible

This is given by the following commands:

Listing 10: Example of a long table

```
\begin{table}[h]
\caption{tabularx table}
\begin{tabularx}{\textwidth}{|l|X|X|}
\hline
5 & some more text is placed here and the space will be divided up equally&
    here we have some more text which will be spread out\\ \hline
8 & We can continue into another column and keep going& This column will also
    be changed to the best size possible\\ \hline
\end{tabularx}
\end{table}
```

6 Figures

6.1 Importing Images

Before placing images within, LaTeX it's important to go over a few formalities. LaTeX only uses JPEG, PNG and GIF file types by default without using other command lines or packages in the preamble. Like with other files used within LaTeX we name the files following the rules that apply to programming languages,

- No spaces within the file name, use underscores.
- Avoid use of special characters.
- Avoid mimicking file extension within file name

As with overleaf files are being uploaded the path of images are determined. However this isn't the case for other applications. In the case that you aren't using Overleaf, the following command is used to specify the path of the folder which has desired images.

```
\graphicspath{ {images/} }
```

It's possible to place images without the need of creating a figure environment (but it can sometimes cause problems), this is done using the following command: `\includegraphics[width=10cm]{astrophysicsrules.jpg}`

6.2 Inserting Figures

Placing figures in LaTeX is quite simple. However, it does require a package, the **graphicx** package. So this must be imported in the preamble. Once this is done you're ready to import and manipulate figures in many ways.

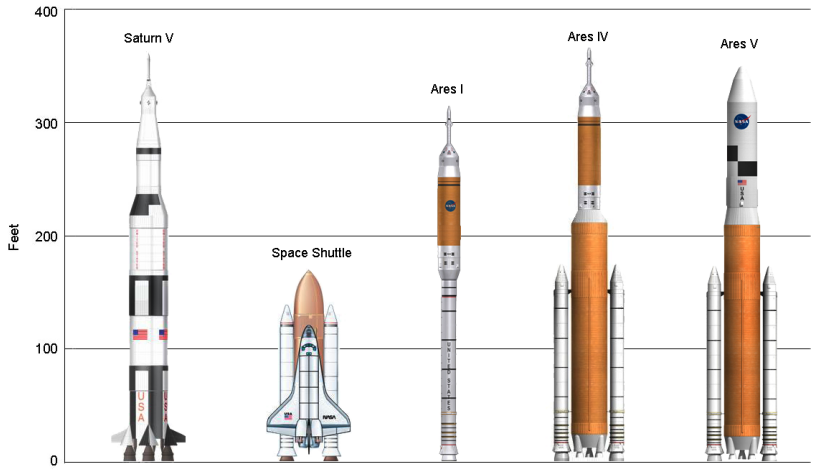


Figure 2: Ares mission craft comparison

Listing 11: Basic example of a single image inserted as a figure

```
\begin{figure}[h]
    \centering
    \includegraphics[width = 11cm]{images/fzLdSPPg.png}
    \caption{Ares mission craft comparison}
    \label{fig:my_label}
\end{figure}
```

The letters in the squared brackets after the initial command line have the following meaning,

These will also work for tables as mentioned before.

Input	Function
t	Top of the page.
r or R	Right side of the page.
l or L	Left side of the page.
h	Will place approximately where indicated in source text if there aren't too many contradictions.
H	Overrides the contradictions stopping *h*, but can cause problems with formatting and compiling.
i or I	Inside the margins on the edge of the page next to the binding on a two page document.
o or O	Outside the margins on the edge of the page opposite to the binding on a two page document.

6.3 Multi-figures

The next useful example is inserting a figure with multiple images,

Listing 12: Example code of a two image figure

```
\begin{figure}
\centering
\begin{subfigure}{.54\textwidth}
  \centering
  \includegraphics[width=1\linewidth]{Pictures/apparatus.png}
  \caption{Apparatus}
  \label{fig:sub1}
\end{subfigure}%
\begin{subfigure}{.4\textwidth}
  \centering
  \includegraphics[width=1\linewidth]{Pictures/pattern.jpg}
  \caption{Pattern on Screen}
  \label{fig:sub2}
\end{subfigure}
\caption{Overview of apparatus throughout experiment.}
\label{fig:test}
\end{figure}
```

(a) Apparatus

(b) Pattern on Screen

Figure 3: Overview of apparatus throughout experiment.

6.4 Wrapping Figures*

It's very common that you may need to wrap text around figures especially in note set or projects. In order to do this another package is needed, the **wrapfig** package. All attributes are defined and edited in the same manner as the previous section. It's important to place wrap figs a moderate distant from the margins of the page including

headers and footers or some funky stuff can happen. [2]

The gravitational assist trajectories at Jupiter were successfully carried out by both Voyagers, and the two spacecraft went on to visit Saturn and its system of moons and rings.
Voyager 1 encountered Saturn in November 1980, with the closest approach on November 12, 1980, when the space probe came within 124,000 kilometers (77,000 mi) of Saturn's cloud-tops.
The space probe's cameras detected complex structures in the rings of Saturn, and its remote sensing instruments studied the atmospheres of Saturn and its giant moon Titan.[33]
Voyager 1 found that about seven percent of the volume of Saturn's upper atmosphere is helium (compared with 11 percent of Jupiter's atmosphere), while almost all the rest is hydrogen. Since Saturn's internal helium abundance was expected to be the same as Jupiter's and the Sun's, the lower abundance of helium in the upper atmosphere may imply that the heavier helium may be slowly sinking through Saturn's hydrogen; that might explain the excess heat that Saturn radiates over energy it receives from the Sun. Winds blow at high speeds in Saturn. Near the equator, the Voyagers measured winds about 500 m/s (1,100 mph).
The wind blows mostly in an easterly direction.[31]

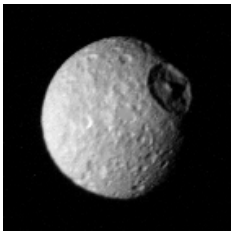


Figure 4: Minmas

Listing 13: Example of warp figure

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \begin{center}
    \includegraphics[width= 4cm]{image.jpg}
  \end{center}
  \caption{Minmas}
\end{wrapfigure}

*TEXT GOES HERE*
```

[2]Figures are a pain to get working at first, it took Hugh and myself hours to get a simple apparatus picture to work on our first lab reports.

7 Equations

7.1 Math Modes

There are two ways in which mathematical notation can be placed within LaTeX. Math mode and the equation function. Math mode allows for equations to be placed within tables and paragraphs, for example $E = \gamma mc^2$. This is done using open and closed dollar signs within a .tex file, as follows `$ E = \gamma mc^2 $`.
The equation function will consider any input between the start and end point as mathematical notation. It will center and automatically label the input by default as follows,

$$E = \gamma mc^2 \tag{1}$$

Where the LaTeX input for the above is as follows,

```
\begin{equation}
    E = \gamma mc^2
\end{equation}
```

The label associated with the equation is dependant on the class parameters of the document. This can be edited using other packages[3]. However it is quite common to use an unnumbered equation. This is done with the use of a asterisk after the equation command as follows `\begin{equation*}  \end{equation*}` .

7.2 Basic Equations

Now that we can enter into math mode the best foundation is to attempt to input some famous equations to get the hang of some of the more arbitrary notation.
As LaTeX interprets spaces as null entries in maths environments a different set of commands are used to manage the spacing. [4]

```
\; - a thick space
\: - a medium space
\, - a thin space
\! - a negative space
```

Alternatively the command `\phantom{---}` can be used to insert blank space. When using this command anything which is typed between the curly brackets will not show as type but as a blank space of the same width.

Basic operators such as +, - and = need nothing special to obtain as they are integrated into modern keyboards, however for things such as indices and fractions a more complicated command is needed.
Fractions
To use fractions within LaTeX the `\frac{ numerator}{ denominator }` command is used, with the numerator and denominator taking respective places. This is better illustrated using a following example,

$$g \; = \; \frac{Gm}{r^2}$$

```
\begin{equation*}
    g \; = \; \frac{Gm}{r^2}
\end{equation*}
```

Indices
Indices are quite simple to use, as in the previous example superscripts are obtained by placing the caret symbol `(^)` after the desired character. For example `$ 4^2 = 16 $` will produce $4^2 = 16$. Similarly subscripts can be obtained by doing the exact same by replacing the caret with a underscore. For example `$ E_i + E_p = E_t$` which

[3] See further readings for details
[4] Each of these different spacing commands have default values associated with them but can be manually changed by redefining them in the preamble

produces $E_i + E_p = E_t$.
These can be combined in many different ways, for example it's possible to obtain this weird expression.

$$Y^{ab}_{\text{initial}} = x^{\sqrt{3}} + z^{\frac{1}{4}}$$

```
\begin{equation*}
    Y_{\text{initial}}^{ab} = x^{\sqrt{3}} + z^{\frac{1}{4}}
\end{equation*}
```

Roots
Roots are obtained using the `\sqrt{}` command, the contents of the curly brackets is the target of the root sign. To give different powers to the root the following is used `\sqrt[x]{}` , which produces $\sqrt[x]{y}$. This is better illustrated in an example,

$$2\pi a^{\frac{3}{2}} \sqrt[\zeta]{\frac{\mu}{B}} = T$$

```
\begin{equation*}
    2 \pi a^{\frac{3}{2}} \; \sqrt[\zeta]{\frac{\mu}{B}} = T
\end{equation*}
```

7.3 Integration

Integration works in a similar way to indices, integrals are initiated using the `\int` command with the limits placed as `\int{lower limit}^{upper limit}`. Thus `$\int_{b}^{a} x^2 \; dx$` produces $\int_b^a x^2 \; dx$.

More complex cases of integrals are as follows,

Double Integrals
`$\iint_V \mu(u,v) \,du\,dv$` produces the following,

$$\iint_V \mu(u,v) \, du \, dv$$

Triple Integrals
`$\iiint_V \mu(u,v,w) \,du\,dv\,dw$` produces the following,

$$\iiint_V \mu(u,v,w) \, du \, dv \, dw$$

Quadruple Integrals
`$\iiiint_V \mu(t,u,v,w) \,dt\,du\,dv\,dw$` produces the following,

$$\iiiint_V \mu(t,u,v,w) \, dt \, du \, dv \, dw$$

Higher Order Integrals
`$\idotsint_V \mu(u_1,\dots,u_k) \,du_1 \dots du_k$` produces the following,

$$\int \cdots \int_V \mu(u_1, \ldots, u_k) \, du_1 \ldots du_k$$

Closed surface integrals
For any closed surface integrals the `esint` package must be used
`$\oint_V f(s) \,ds$` produces,

$$\oint_V f(s) \, ds$$

and, `\oiint_V f(s,t) \,ds\,dt` produces,

$$\oiint_V f(s,t) \, ds \, dt$$

7.4 Sums & Limits

Both sums and limits work on the same basis as integration in the previous section. For sums the basic command is as follows, `\sum_{Lower}^{Upper}`, which when compiled by math mode produces,

$$\sum_{n}^{\infty} f(x)$$

The product of a sequence of numbers is done in the same way, however the sum is replaced with prod as follows. `\prod_{lower}^{upper}` which then produces the following,

$$\prod_{n}^{\infty} f(x)$$

Limits are then produced within the math environment using the following command, `\lim_{n\to\infinity}` which when run in math mode produces,

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

Using all above methods in a complex Taylor series at $x = 0$ will help show how this is useful tool to use within documents,

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f(0)}{2!}x^2 + \frac{f'(0)}{3!}x^3 + \cdots = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!}x^k$$

Listing 14: Code used to produce above equation

```
f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f(0)}{2!}x^2 +
\frac{f'(0)}{3!}x^3+\dotsb =
\sum_{k=0}^\infty \frac{f^{\left(k\right)}(0)}{k!} x^k
```

7.5 Arrays

For maths that need to be inputted over multiple lines it can be done using an array[5]
For example we have,

$$\begin{aligned}
&\lim_{x\to -2} \frac{x^3 - 8}{x^2 + 4} \\
=&\frac{-2^3 - 8}{-2^2 + 4} \\
=&\frac{-16}{8} \\
=&-2
\end{aligned}$$

This is then inputted using the following block of code.

```
\begin{equation*}
\begin{array}
& \lim_{x \to -2} \frac{x^3 - 8}{x^2 + 4} \\
= & \frac{-2^3 - 8}{-2^2 + 4} \\
= & \frac{-16}{8} \\
= & -2
\end{array}
\end{equation*}
```

[5]Other ways of doing this, but arrays are for the most part the most convenient way.

7.6 Matrices

Matrices operate in the same method as tables. There are many different brackets and forms that matrices can take in LaTeX, in this workshop we will just look at three of the most common matrices forms that appear.
Matrix without any containing notation

$$\begin{matrix} 1 & 3 & 3 & 4 \\ 2\eta & \alpha & \gamma & \beta \end{matrix}$$

```
\begin{center}
\begin{equation*}
\begin{matrix}
1 & 3 & 3 & 4 \\
2 \eta & \alpha & \gamma & \beta
\end{matrix}
\end{equation*}
\end{center}
```

Matrix without contained with round brackets,

$$\begin{pmatrix} 1 & 3 & 3 & 4 \\ 2\eta & \alpha & \gamma & \beta \end{pmatrix}$$

```
\begin{center}
\begin{equation*}
\begin{pmatrix}
*matrix input*
\end{pmatrix}
\end{equation*}
\end{center}
```

Matrix without contained with square brackets,

$$\begin{bmatrix} 1 & 3 & 3 & 4 \\ 2\eta & \alpha & \gamma & \beta \end{bmatrix}$$

```
\begin{center}
\begin{equation*}
\begin{bmatrix}
*matrix input*
\end{bmatrix}
\end{equation*}
\end{center}
```

Matrix without contained with curly brackets,

$$\begin{Bmatrix} 1 & 3 & 3 & 4 \\ 2\eta & \alpha & \omega & \beta \end{Bmatrix}$$

```
\begin{center}
\begin{equation*}
\begin{Bmatrix}
*matrix input*
\end{Bmatrix}
\end{equation*}
\end{center}
```

7.7 Using SI units & Scientific notation

Due to the nature of physics lab reports it's quite common to come across data that needs to be interpreted in scientific notation, it's very possible to handle this using tools in the **amsmath** package (few of us do just use the base package), it can be streamlined using the **siunitx** package, which imports plenty of useful tools when it comes to writing lab reports.
To use scientific notation the following command is used within math mode `\num{2.99e8}`. Which produces 2.99×10^8. SIUnitx has many really useful features which can be further explored in the manual which is linked in the further reading section, unfortunately due to this workshop giving just the basics of LaTeX this material is OuT OF ThE ScOPe Of THiS wOrKshoP.

7.8 Other useful techniques*

As this is an introduction workshop the full capabilities of LaTeX will not be explored, however this section includes some useful notation that was required during our previous classes.
Encasing equations in brackets
Braces in LaTeX

$$E_u = \underbrace{\frac{-\hbar^2}{2m}\left(\frac{\partial^2 U}{\partial x^2}\right)}_{\text{Kinetic Energy}} + \overbrace{\frac{2m(V_0 - E)U}{\hbar^2}}^{\text{Potential Energy}}$$

Braces are implemented in a similar fashion to limits and integrals, seen as follows,

```
\begin{equation*}
 E_u =  \underbrace{equation}_{\text{Kinetic Energy}} + \overbrace{equation}^{\
    text{Potential Energy}}
\end{equation*}
```

Cases in LaTeX

$$V(x) = \begin{cases} V_0 & x = -\frac{L}{2} \\ 0 & -\frac{L}{2} \leq x \leq \frac{L}{2} \\ V_0 & x > \frac{L}{2} \end{cases}$$

```
\begin{equation*}
   V(x) =
    \begin{cases}
        V_0 & x = -\frac{L}{2} \\
        0 & - \frac{L}{2} \leq x \leq \frac{L}{2} \\
        V_0 & x > \frac{L}{2}
    \end{cases}
\end{equation*}
```

7.9 Other mathematical symbols

LaTeX is able to type many maths symbols that we wont mention here but to see a list of the most commonly used ones see the further reading section and look at point 9.

8 Placing code within a LaTeX document

In the case of Lab Reports the code used to graph and process data needs to be provided in the appendices. To do this a special environment needs to be implemented, this can be done using two packages. The **listings** package and the **verbatim** package.

8.1 Verbatim

This environment will place code within document. To place within a normal string of text `\verb` with a pair of modulus after the command, with the code placed within the boundaries of the modulus. It's also possible to use verbatim by initiating an environment much like math mode (equations) and figures. For example,

```
\begin{verbatim}

    *code input*

\end{verbatim}
```

8.2 Listings

Listings are much like the previous section, the main difference being how each package works and the aesthetics that can be altered. Listings use the **listings** package.
The code displayed in the boxes throughout this document are listings. The listing environment is initiated in the same way as verbatim instead using *lstlisting* instead of *verbatim*.
This is better illustrated using an example.

Listing 15: This is a sample listing

```
\begin{lstlisting}[frame = single, caption = {\text{This is a sample listing}}]

    *code input*

\end {lstlisting}
```

The parameters of listings can be edited to change the appearance of the code. See further reading for more details on this.

9 Defining Commands

One of the most useful things to do in LaTeX is defining commands to improve quality of life while creating projects. Commands can range from hot-keys to in placing a complex graphic with a simple commands. For example, Joe uses the following commands to create hotkeys that allow for fast input of math equations during note writing.

Listing 16: Example hotkeys

```
\newcommand{\newindent}{\newline \indent}
\newcommand{\sus}{\subsection}
\newcommand{\m}{\boldsymbol}
\newcommand{\be}{\begin{equation*}}
\newcommand{\ee}{\end{equation*}}
\newcommand{\sub}{\subsubsection*}
\newcommand{\new}{\newline\newline}
\newcommand{\lgap}{\hspace{1cm}}
\newcommand{\sgap}{\hspace{0.1cm}}
```

As seen above the `\newcommand{name}{function}` defines a function that is accessed by the name and executes the function within the respective round brackets. Now if Joe needs to begin an equation the command `be` is used instead of `\begin{equation*}`.
7
Another example of this is defining commands to shorten aspects of a tedious input that comes up frequently whilst working. An example of this is vector notation and hats on variables. Prof. Duffy defines the following commands as macros for things that appear frequently in note writing as seen in the below example.

```
\newcommand{\ihat}{\hat{\bf i}}
\newcommand{\jhat}{\hat{\bf j}}
\newcommand{\khat}{\hat{\bf k}}
```

Thus, $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ is produced in a single short command.

10 Referencing

10.1 Labelling Equations and Figures

During projects you often have to refer back to tables, equations and figures. If you label and reference these items LaTeX will automatically handle any changes in the order of numbers. This way there is no need to go back and manually replace the numbers throughout the project. This is done using the following commands. `\label{title}` is used to initially label it and `\ref{title}`

```
Please see Fig. (\ref{title})
```

LaTeX has a number of packages that nicely handle referencing within projects. **BibTeX** is the oldest and most common package used.

10.2 Creating a .bib file

BibTeX file contains all the references you want to cite in your document. It is created in the same manner as a .tex file however it has the **.bib** extension. It is placed in the same directory as the main.tex[6]. The file is where all the references are placed in a format as seen in the below listings.

Listing 17: Manual example

```
@MANUAL{PHYC20020,
    title        = "PHYC20020: Quantum Mechanics Lab Manual",
    author       = "UCD Department Of Physics",
    organization = "University College Dublin",
    address      = "UCD, Belfield",
    month        = "September",
    year         = "2019"
}
```

Listing 18: Article example

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

These can be entered into the .bib file manually but it can be streamlined using automatic citing websites such as `http://www.citationmachine.net/bibtex/`. Websites will automatically find most sources out of a large indexed data base and automatically input information into the fields as seen in the previous listings. It's **important** to note that these automatic citing websites sometimes miss clinical information or mis-input information. Although convenient it's always important to check that your references are appropriately formatted displaying the correct information as the S.T.E.M field does not exist without collaboration.

10.3 Placing bibliography within document

Like with the table of contents the bibliography is placed in the document with a single command with excess parameters to customize aspects of the bibliography. To place in the document the following command is used `\bibliography{filename}`. The rounded brackets is the name of the .bib file excluding the .bib extension.

[6] Unless directory specified in preamble

10.4 Referencing within a document

To reference within the document using the following command is used, `\cite{label}`. Where the label is the title given to the reference in the .bib file. In the case of the examples above the labels are **PHYC20020** and **Birdetal2001**. For example if an apparatus picture is taken from a lab manual it would be referenced within the document as,

Listing 19: Example figure caption with reference

```
\caption{Configured apparatus throughout the experiment \cite{PHYC20020}}
```

10.5 Styles

Depending on personal preference and the type of document in question different referencing styles may need be used. BibTeX does a good job of changing the referencing style of the fly. To change the style of the bibliography the following command is used, `\bibliographystyle{plain}` where the entry within the round bracket is used as the parameter to define the style. A common style used is **APA** and **ACM** see further reading to view further styles and examples.

11 Bug fixing and Trouble shooting

As with any coding language LaTeX will produce an error if the code is not fully correct also as we are specifying every aspect of the document this will occur often. The compiler will attempt to correct any mistakes in the code, however, if there is a large error it might be unable to compile the document. When any error occurs an error message will appear and it will try to tell you what the issue is. Most compilers will also highlight which line of code in particular is causing the issue. Sometimes these error messages may not be entirely clear as the compiler may not be able to determine what the exact problem is or what you are trying to get it to do. When working in LaTeX it is important to compile your code regularly to check that the document is rendering in the way you want. It is also important to compile it regularly as if you don't and you get an error when running a large amount of code it can be difficult to figure out exactly where the problem is.

The following are some common errors to check for if your code is producing an error.

- A missing bracket `{}`. All brackets must be closed
- An open environment. If you begin an environment with `\begin{}` you must also close it using `\end{}`.
- if you are typing maths equations in text you must open and close it with $.

If your problem is an error like one of the ones above it is often a good idea to remove the code which is causing the problem and to reinsert it section by section again and see which part of it is causing the problem.
Unfortunately, most of the problems you encounter will probably be more complicated than this or there may be multiple mistakes in your code. Also there are an infinite number of errors so we can't give you a definitive list of all of the things to check. Many other people have already encountered most of these problem and so there are solutions to most issues online. Simply google your problem and you will more than likely find a solution. If you are unsure what to even ask google (this happened us a lot when we started learning LaTeX!) try typing in the name of the error message that your compiler gives you.

Debugging any code in LaTeX can be incredibly frustrating but if you stick with it, it does become easier!

Further Reading

1. **LaTeX Compilers**:
   https://en.wikipedia.org/wiki/Comparison_of_TeX_editors
2. **Packages**: Extra reading and information on packages.
   https://en.wikibooks.org/wiki/LaTeX/Installing_Extra_Packages
3. **Table of Contents**: How to further change the style of TOC.
   https://texblog.org/2011/09/09/10-ways-to-customize-tocloflot/
4. **CSVsimple Import Manual**:
   https://osl.ugr.es/CTAN/macros/latex/contrib/csvsimple/csvsimple.pdf
5. **Labelling Equations**:
   https://support.authorea.com/en-us/article/how-do-i-reference-my-latex-tables-or-equations-brkb5m/
6. **SIUnitx Package Manual**:
   http://tug.ctan.org/macros/latex/exptl/siunitx/siunitx.pdf
7. **Listings Manual**:
   http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf
8. **Referencing Styles**:
   https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management
   https://www.overleaf.com/learn/latex/Biblatex_citation_styles
9. **Consise list of maths operators and symbols:**
   http://tug.ctan.org/info/undergradmath/undergradmath.pdf
10. **Very Comprehensive List of Symbols**:
    http://ctan.math.illinois.edu/info/symbols/comprehensive/symbols-a4.pdf
11. **Overleaf's LaTeX guides:**
    https://www.overleaf.com/learn
12. **LaTeX Minipages:**
    https://www.sascha-frank.com/latex-minipage.html
13. **Workshop LaTeX Source Code:**
    https://pastebin.com/r8cwZuVw

[7]

[7] This note set is about as long as a lab report :(