Honours LaTeX workshop

1 Introduction
- Typesetting mathematics
- Modes

1. Introduction
   - Typesetting mathematics
   - Modes
2. Text typesetting
   - Spacing
   - Lists
   - Tables

1. Introduction
   - Typesetting mathematics
   - Modes
2. Text typesetting
   - Spacing
   - Lists
   - Tables
3. Math typesetting
   - Numbered equations
   - Labels and referencing
   - Multiple equations

1. Introduction
   - Typesetting mathematics
   - Modes
2. Text typesetting
   - Spacing
   - Lists
   - Tables
3. Math typesetting
   - Numbered equations
   - Labels and referencing
   - Multiple equations
4. Advanced topics

1. Introduction
   - Typesetting mathematics
   - Modes
2. Text typesetting
   - Spacing
   - Lists
   - Tables
3. Math typesetting
   - Numbered equations
   - Labels and referencing
   - Multiple equations
4. Advanced topics
5. Advanced math typesetting

1. Introduction
   - Typesetting mathematics
   - Modes
2. Text typesetting
   - Spacing
   - Lists
   - Tables
3. Math typesetting
   - Numbered equations
   - Labels and referencing
   - Multiple equations
4. Advanced topics
5. Advanced math typesetting
6. Advanced text typesetting

What is LaTeX?

- "LaTeX is a document preparation system for high-quality typesetting."[1]
- "It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing."[2]
- LaTeX is an extension of TeX which was "designed and mostly written by Donald Knuth and released in 1978."[3]
- LaTeX is programmable (in essence a programming language of its own) and very powerful to use once the basics have been mastered.
- This presentation was typeset in LaTeX.

[1] http://latex-project.org/intro.html [Accessed 28 February 2014]
[2] http://latex-project.org/intro.html [Accessed 28 February 2014]
[3] http://en.wikipedia.org/wiki/TeX [Accessed 28 February 2014]

Introduction

Example

The following *source code*:

```
\documentclass[a4paper,12pt]{article}
\usepackage{amsmath,amsfonts,amssymb}
\begin{document}
  Hello world!
\end{document}
```

would produce a document with only the line "Hello world!" in it.

- Try this now in your LaTeX IDE or text editor (IDE stands for Integrated Development Environment).
- Save the file as `myfile.tex` and compile it.
- Try replacing the text "Hello world!" with a sentence of your own making.

Introduction (cont.)

- The `\documentclass[10pt]{article}` tells LaTeX that you want to produce an `article` with paper size A4 and font size of 12pt (default font size is 10pt). There are other classes like `book` and `report` and other options for the `\documentclass` command.
- `\usepackage{amsmath,amsfonts,amssymb}` that we want to make use of the named packages. Packages are a collection of useful commands and are extensions of LaTeX.
- The commands `\begin{document}...\end{document}` define an *environment* and will contain everything we want to typeset.
- It is important to have the commands `\documentclass{...}` and `\begin{document}...\end{document}` in every LaTeX file you create, otherwise you shall get errors when you try to compile your files.

Typesetting mathematics

Example

Insert the following

```
$f(x) = x$
```

between the words “Hello” and “world!” in your `myfile.tex` file.

The result should look like this:

Output

Hello $f(x) = x$ world!

Typesetting mathematics (cont.)

- We use the delimiters `$...$` to typeset mathematics *inline*.
- What if we wanted to display a formula on its own?
- Answer: we use the delimiters `\[...\]`.

Example

Press `Enter` twice after your last sentence in your file `myfile.tex`. Next enter the following code:

```
\[
  f(x) = x
\]

The above equation is an example of
a display-type equation.
```

and compile/build your file again.

Typesetting mathematics (cont.)

The output of your previous code should be the following:

Output

Hello $f(x) = x$ world!

$$f(x) = x$$

The above equation is an example of a display-type equation.

Modes

- There is a difference when you typeset text and mathematics in LaTeX.
- The default in LaTeX is text mode.
- Once you use the delimiters `$...$` or `\[...\]` you enter mathematics typesetting mode.
- The distinction is important!

Example

At the end of your `myfile.tex` enter the following code:

```
\frac{x}{n}
```

and compile your file.

Modes (cont.)

- What was the output of your compilation?
- Did you also get an error similar to "Missing $ inserted."?
- This happened because we were trying to use a command in text mode which can solely be used in math mode.
- Change `\frac{x}{n}` to either `$\frac{x}{n}$` or `\[\frac{x}{n}\]` and compile the code again.
- Your code should now compile without any problem.
- You should see either $\frac{x}{n}$ or $\frac{x}{n}$.

Commands

- We aren't just restricted to regular text and the delimiters to typeset mathematics.
- LaTeX puts a wide variety of commands at your disposal to typeset mathematics and text.
- In LaTeX commands start with the backslash ('\') character and take the form

$$\backslash command[options]\{parameters\}$$

- For example, the command `\emph{...}` emphasizes text by making it italic. So `\emph{This is emphasized}` would produce "*This is emphasized*".

Comments

- Since commands make LaTeX a programming language, we should also have the ability to comment out commands.
- This is done by % symbol.
- Anything following a % symbol is not interpreted by LaTeX when you compile your documents.
- However, note that a comment is only per line.
- So if you want a multi-line comment you must put a % symbol on each line.

Spacing

- LaTeX ignores most white space.
- Try inputting four or five spaces between sentences and compiling your example document.
- You should see that LaTeXcompresses this to a single space between the sentences in your output document.

Paragraphs

Input

```
This is a sentence to illustrate how \LaTeX\
handles paragraphs.

This is the first sentence in the new
paragraph.
```

Output

This is a sentence to illustrate how LaTeX handles paragraphs.
This is the first sentence in the new paragraph.

Lists

LaTeX makes lists easy as well. The following code shows how to produce a bullet list:

Input

```
\begin{itemize}
  \item Item 1
  \item Item 2
  \item Item 3
\end{itemize}
```

Output

- Item 1
- Item 2
- Item 3

Lists (cont.)

The following code shows how we produce a numbered list:

Input

```
\begin{enumerate}
  \item First item
  \item Second item
  \item Third item
\end{enumerate}
```

Output

1. First item
2. Second item
3. Third item

We can even create lists within lists! We leave it up to you to experiment.

Tables

We can create tables by using the `table` and `tabular` environments. For example, the following table

Time	Men	Women	Time
10:00	4	5	10:00
12:00	8	3	12:00

is produced by the code

```
\begin{table}
  \begin{tabular}{|lccr|}
    \hline
    \textbf{Time} & \textbf{Men} & \textbf{Women} &
    \textbf{Time} \\
    \hline
    10:00 & 4 & 5 & 10:00 \\
    12:00 & 8 & 3 & 12:00 \\
    \hline
  \end{tabular}
\end{table}
```

Tables (cont.)

- The `tabular` environment is the bare-bones basic you need to produce a table and can be used without the `table` environment.
- The part `{|lccr|}` next to the `\begin{tabular}` command shows that we want four columns in the table with a vertical line at the left and right edge of the table. The `l`, `c` and `r` letters indicate the alignment we want within the column and stand for left, center and right respectively.
- Following the `\begin{tabular}{|lccr|}` we have the actual rows of our table and their entries.
- The ampersand & is used to separate the different columns.
- The double backslash \\ terminates the row and goes on to the next line in the table.
- The command `\hline` draws a horizontal line for the full width of the table.

Math typesetting recap

- We saw previously that we could typeset mathematics.
- For in-line equations use the `$...$` delimiter.
- To display the equation on its own we use the `\[...\]` delimiter.

Numbered equations

- LaTeX has the ability to produce numbered equations, which we can label and reference at a later stage.
- This is done by using the `equation` environment, i.e. `\begin{equation}...\end{equation}`

Input

Insert the following code into your `myfile.tex` file:

```
\begin{equation}
  f(x) = x
\end{equation}
```

Numbered equations (cont.)

- You should have the following output:

Output

$$f(x) = x \tag{1}$$

- LaTeX automatically inserted the equation number on the right and increases it with subsequent use of `equation` environments.

Input

Insert the following immediately after the previous equation:

```
\begin{equation}
  f(x)=x+1
\end{equation}
```

Numbered equations (cont.)

You should have the following output:

Output

$$f(x) = x \tag{1}$$

$$f(x) = x + 1 \tag{2}$$

Labels

LaTeX provides the functionality to label equations, sections, chapters, and so forth, and reference at a later stage by using the `\label{...}` and `\ref{...}` commands.

Example

Change the numbered equations you previously typeset as follows:

```
\begin{equation}\label{first-equation}
  f(x)=x
\end{equation}

\begin{equation}\label{second-equation}
  f(x)=x+1
\end{equation}
```

You can use whatever you want as the labels within the label brackets. Compile your document *twice*.

Referencing

Your output should look the same as it did previously. The reason is that labelling equations is an internal process in LaTeX and you only see the value of it once you reference your labelled equations.

Example

Add the following line after your numbered equations:

```
We can reference the equations \ref{first-equation}
and \ref{second-equation}.
```

and compile your document again.

Referencing (cont.)

You should now see something like the following in your output:

Output

$$f(x) = x \tag{1}$$

$$f(x) = x + 1 \tag{2}$$

We can reference the equations 1 and 2.

Aligning equations

- What if we wanted the equal signs of your two equations aligned?
- LaTeX has an `align` environment which will be useful.

Example output

$$g(y) = y$$
$$h(y) = y + 1$$

The equations above were produced by the following code:

Example input

```
\begin{align*}
  g(y) &= y \\
  h(y) &= y+1
\end{align*}
```

Aligning equations

- The `align` environment looks similar to a table in LaTeX.
- Notice that we did not need to use any delimiters to enter the equations.
- Reason being that the `align` environment is in math mode by default.

More mathematics typesetting

- We had fun with the basic ideas of typesetting in LaTeX - we covered basic mathematics and text typesetting.
- This time around we look at some "advanced" concepts.

Subscripts and superscripts

- What if we want to add superscripts (powers) and subscripts (indexes) to variables in equations?
- In LaTeX we use an underscore (_) for subscripts/indices and a caret (^) for superscripts/powers.

Type the following in *math* mode (display or inline—your choice) in your LaTeX file:

```
x_2 x^3
```

to produce

$$x_2 x^3$$

Subscripts and superscripts (cont.)

- Now typeset either `x_23` or `x^23`.
- Your output should looks something like "x_23" or "x^23".
- LaTeX only lowers or raises the expression immediately after the underscore or caret.
- For multiple expressions, we have to make use of curly braces { }.
- Change the previous expressions to `x_{23}` or `x^{23}` and compile your file.
- The output should now look like "x_{23}" or "x^{23}".
- We can combine subscripts and superscripts, e.g. `x_{ij}^{23}` produces "x_{ij}^{23}".

Symbols

- There are certain symbols which can be directly entered from the keyboard:

$$+ \quad - \quad = \quad ! \quad / \quad (\quad) \quad [\quad] \quad < \quad > \quad | \quad ' \quad ;$$

- Beyond these symbols listed above, we need distinct commands to display the desired symbols.

Type the following in *math* mode (display or inline—your choice) in your LaTeX file:

```
\forall x\in X, \quad \exists y \leq \epsilon
```

to produce

$$\forall x \in X, \quad \exists y \leq \epsilon$$

Greek symbols

- It should come as no surprise that LaTeX allows us to typeset the Greek alphabet, which we commonly use for variables in mathematics.
- They are easy to type in *math* mode, and done by typing the Greek letter's name as a command.

Type the following in *math* mode in your LaTeX file

```
\alpha, A, \beta, B, \gamma, \Gamma, \pi, \Pi
```

to produce

$$\alpha, A, \beta, B, \gamma, \Gamma, \pi, \Pi$$

Operators

- Operators are LaTeX commands which is written as a word, e.g. trigonometric functions, logarithms and exponentials.
- For example, `\cos(x)` produces $\cos(x)$.

Typeset the following in your file

```
\sin^2 \theta + \cos^2 \theta = 1
```

to produce "$\sin^2 \theta + \cos^2 \theta = 1$".

- Certain operators, like the limit (`\lim`) of a function, require extra information. For the limit, we have to specify a subscript.

Typeset the following in your file

```
\lim_{n\rightarrow\infty}\exp(-x)=0| produces
```

to produce "$\lim_{n\rightarrow\infty} \exp(-x) = 0$".

Fractions and roots

- To typeset fractions in LaTeX we make use of the `frac` command.
- `\frac` takes two arguments—the numerator and denominator.

Enter the following into your file

```
\frac{x+1}{x}
```

to produce

$$\frac{x+1}{x}$$

- Square and n-roots are typeset using the `\sqrt{..}` command. For example, `\sqrt{2}` produces $\sqrt{2}$.
- If we give `\sqrt` a number option (in square brackets before the curly brackets), it produces that numbered root. For example, `\sqrt[3]{2}` produces $\sqrt[3]{2}$.

Sums and integrals

- Sums (`sum`) and integrals (`int`) follow the same pattern when we typeset them in LaTeX
- We need to supply both with a subscript and superscript.

Enter the following into your file

```
\sum_{k=1}^{\infty} \frac{1}{x^k}
```

to produce

$$\sum_{k=1}^{\infty} \frac{1}{x^k}$$

Enter the following into your file

```
\int_a^b \frac{1}{x}dx=\ln(x)
```

to produce

$$\int_a^b \frac{1}{x}dx = \ln(x)$$

Matrices and arrays

- The most basic way of typesetting matrices and arrays is to use the `matrix` environment.

Type the following into your program, preferably as a display type:

```
\begin{matrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{matrix}
```

to produce

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

Matrices and arrays (cont.)

- When we write mathematics, we use brackets as delimiters for the matrix, either [ ] or ( ).

Modify your matrix, by putting either [ ] or ( ) before and after your `matrix` environment commands. For example

```
[\begin{matrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{matrix}]
```

will produce

$$\begin{matrix}
  a & b & c \\
  [d & e & f] \\
  g & h & i
\end{matrix}$$

- How do we get the delimiters to cover the complete sides of the matrix?

Matrices and arrays (cont.)

- When use the `left` and `right` commands.

Modify your matrix, by putting either `\left[ \right]` or `\left( \right)` before and after your `matrix` environment commands. For example

```
\left[\begin{matrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{matrix}\right]
```

will produce

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

- **Important!** `\left` and `\right` must always be paired up, otherwise you will incur errors at compile time!

Matrices and arrays (cont.)

By using the `amsmath` or `mathtools` packages, we can make use of modified `matrix` environments so they automatically include delimiters.

Environment	Delimiter	Notes
`pmatrix`	$(\ )$	Centers columns
`pmatrix*`	$(\ )$	Alignment may be specified with optional parameter
`bmatrix`	$[\ ]$	Centers columns
`bmatrix*`	$[\ ]$	Alignment may be specified with optional parameter
`Bmatrix`	$\{\ \}$	Centers columns
`Bmatrix*`	$\{\ \}$	Alignment may be specified with optional parameter

Matrices and arrays (cont.)

Environment	Delimiter	Notes
`vmatrix`	$\vert\ \vert$	Centers columns
`vmatrix*`	$\vert\ \vert$	Alignment may be specified with optional parameter
`Vmatrix`	$\Vert\ \Vert$	Centers columns
`Vmatrix*`	$\Vert\ \Vert$	Alignment may be specified with optional parameter

- The starred environments require the `mathtools` package to be included in your document preamble.
- The non-starred environments require the `amsmath` package to be include in your document preamble.

Subordinate equation numbering

Consider the following equations:

$$\dot{x} = p_x + y \tag{3a}$$
$$\dot{y} = p_y - x \tag{3b}$$
$$\dot{p}_x = p_y + \frac{\partial \Omega}{\partial x} \tag{3c}$$
$$\dot{p}_y = -p_x + \frac{\partial \Omega}{\partial y} \tag{3d}$$

Equations (3) describe the motion of a satellite in the planar, circular, restricted three-body problem. Equations (3a) and (3b) will be solved for the x and y coordinates, while equations (3c) and (3d) will be solved for the corresponding conjugate momenta.

Subordinate equation numbering (cont.)

The previous frame was typeset from the following

```
Consider the following equations:
\begin{subequations}\label{eqn:pcrtbp}
  \begin{align}
    \dot{x} &= p_x + y \label{eqn:pcrtbp-xcoord} \\
    \dot{y} &= p_y - x \label{eqn:pcrtbp-ycoord} \\
    \dot{p}_x &= p_y + \frac{\partial \Omega}{\partial x}
          \label{eqn:pcrtbp-xmomenta} \\
    \dot{p}_y &= -p_x + \frac{\partial \Omega}{\partial y}
          \label{eqn:pcrtbp-ymomenta}
  \end{align}
\end{subequations}
Equations \eqref{eqn:pcrtbp} describe the motion of
a satellite in the planar, circular, restricted three-body
problem.
Equations \eqref{eqn:pcrtbp-xcoord} and
\eqref{eqn:pcrtbp-ycoord} will be solved for the $x$ and $y$
coordinates,
```

Paragraph alignments

We can change the way LaTeX aligns text.

We make use of the `center`, `flushleft` and `flushright` environments.

We may also make use of the commands `\centering`, `\raggedright` and `\raggedleft` for these environments, but they may lead to improper typesetting if used wrongly!

Sectioning a document

- Inserting sections into our documents is very easy in LaTeX, we make use of the `\section{..}` command.
- Different document classes allow different levels of sectioning.
- For example, the `book` class allows chapters via the `chapter` command and parts via the `part` command, but the article class would not allow these.
- The level in sectioning is as follows

Command	Level	Notes
`\part{..}`	-1	May not be used in letters
`\chapter{..}`	0	only in books and reports
`\section{..}`	1	May not be used in letters
`\subsection{..}`	2	May not be used in letters
`\subsubsection{..}`	3	May not be used in letters
`\paragraph{..}`	4	May not be used in letters
`\subparagraph{..}`	5	May not be used in letters

Sectioning (cont.)

Try inserting the following into your file:

```
\newpage
\section{Header of the first section}
\subsection{Header of the first subsection}
\section{Header of the second section}
```

Your output should list the sentences between curly braces in bold with numbers to the left of the sentence. LaTeX automatically numbers sections for us as well!

Fonts

- We know that the `\emph{..}` command emphasizes text. LaTeX gives us the ability to change the typesetting of fonts as we need them.
- To get a sentence or part of it to be bold, we make use of the `\textbf{..}` command. **This sentence is an example of that**.
- To get a sentence or part of it to be italic, we make use of the `\textit{..}` or `\emph{..}` commands. *This sentence is an example of that*.
- The `\texttt{..}` command turns the text into teletype font, which is a fixed-width or monospace font. `It is especially used when typesetting computer programming code.`

Fonts (cont.)

- We can change the size of fonts.

Type the following into your document:

```
sample text \\
{\small sample text} \\
\Large{sample text}
```

to produce

sample text
sample text
sample text

See the difference?

Conclusion

- We have merely touched the tip of the iceberg when it comes to LaTeX.
- The documentation we have provided you will be a helpful place to start when you get stuck somewhere.
- Of course Google is your friend, there are huge amounts of information and symbol lists and help available online.