

# MU $\text{\LaTeX}$ Workshop

September 2021



Introduction

Basic Writing

Formatting Documents

Mathematics

Alignment and Arrays

Figures

Concluding Remarks

# Introduction: What is $\text{\LaTeX}$ ?

- ▶  $\text{\LaTeX}$  uses markup convention opposed to WYSIWYG editors
- ▶ Allows for consistent and scalable styling while you focus on writing the content.
- ▶ You tell  $\text{\LaTeX}$  what it is, not what it should look like.
- ▶ We will use *Overleaf* today, an online  $\text{\LaTeX}$  editor

# Introduction: How Does it Work?

- ▶ Prepare a `main.tex` file
- ▶ Use markup commands to tell  $\text{\LaTeX}$  what to do.
- ▶  $\text{\LaTeX}$  reads your commands and compiles the document.
- ▶ Produces a `main.pdf` file

## Example 1

```
\begin{equation}  
  \sum_{i=1}^{k+1} n^2  
\end{equation}
```

$$\sum_{i=1}^{k+1} n^2 \quad (1)$$

# Introduction: Master Keys

- ▶ Backslash `\` is the master key - distinguishes command from content.
- ▶ Curly brackets `{}` are used to contain and define commands
- ▶ Percent sign `%` is used for comments
- ▶ Other special characters include
  - ▶ `#`
  - ▶ `&`
  - ▶ `—`
  - ▶ `^` (caret)

# Introduction: Macros and Environments

- ▶ Commands are either macros or environments
- ▶ Macros begin with \
- ▶ Environments are surrounded by \begin{environment} and \end{environment}
- ▶ Commands may sometime require *arguments* given using curly brackets {} or square brackets []

## Example 2

```
\[  
\left|  
\begin{array}{cc}  
2-\lambda & 0\\  
4 & 3-\lambda\end{array}  
\end{array}  
\right|  
\]
```

$$\begin{vmatrix} 2-\lambda & 0 \\ 4 & 3-\lambda \end{vmatrix}$$

# Introduction: Document Set Up

## Typical Document Set Up

```
\documentclass[a4paper,14pt]{article}
\usepackage[utf8]{inputenc}

\title{ Title }
\author{ Author }
\date{ Date }

\begin{document}

\maketitle

\section{Introduction}
Text Here!
\end{document}
```

# Introduction: Packages

They are just macros and environments other people have written!  
Some of the most useful to use:

Package	Use
amsmath, amssymb, amsfonts	various maths tools
geometry	to finely tune page size + margins
pgfplot	to create figures and plots
hyperref	to embed links
graphicx	to add graphics and figures
siunitsx	for units in maths



Introduction

Basic Writing

Formatting Documents

Mathematics

Alignment and Arrays

Figures

Concluding Remarks

# Basic Writing: Text Options

We can change font options either using the

## 1. **command**

- ▶ Surround small blocks of text
- ▶ `\command{text}`

## 2. **switch command**

- ▶ Convert large blocks of text
- ▶ Use `\switchcommand`
- ▶ Will keep text option unless changed or;
- ▶ can surround text with `{\switchcommand text}`

# Basic Writing: Text Options

Can change font size:

<code>\scriptsize</code>	Example
<code>\small</code>	Example
<code>\large</code>	Example
<code>\Large</code>	Example
<code>\LARGE</code>	Example
<code>\huge</code>	Example
<code>\Huge</code>	Example

Each can act as a switch command or normal command.

# Basic Writing: Text Options

Can change font style:

<code>\textbf</code>	<b>Example</b>	Bold
<code>\textit</code>	<i>Example</i>	Italic
<code>\textsc</code>	EXAMPLE	Small Caps

Or font family:

<code>\textrm</code>	Example	Roman
<code>\texttt</code>	Example	Typewriter
<code>\textsf</code>	Example	Sans Serif

Whose switch commands can be evoked using `\rmfamily` etc.

# Basic Writing: Text Options

Can change font colour:

## Changing Text Colour

Here is how you can change the colour of single `\textcolor{red}{words}` using the command, or changing `{ \color{red}` blocks of texts by surrounding it in curly brackets `}` using the switch command.

Here is how you can change the colour of single **words** using the command, or changing **blocks of texts by surrounding it in curly brackets** using the switch command.

The package `xcolor` is particularly useful for getting very specific colours.

# Basic Writing: Lists

- ▶ We can make ordered and unordered lists using the following environments:
  1. `\begin{enumerate}` for unordered lists
  2. `\begin{itemize}` for ordered lists
  3. `\begin{description}` for long paragraphs
- ▶ Each point is denoted using the `\item` command
- ▶ You can nest up to four lists
- :) You can change the bullet by specifying it as an argument in `\item[new bullet]`

# Basic Writing: Lists

## Example List

```
\begin{itemize}
  \item Heres an example of a list!
    \begin{enumerate}
      \item And how you can nest lists!
      \item Of any kind!
    \end{enumerate}
  \item[!] And how you can change bullet points
\end{itemize}
```

Introduction

Basic Writing

Formatting Documents

Mathematics

Alignment and Arrays

Figures

Concluding Remarks



# Formatting Documents: Title Pages

- ▶ We can make a title with `\maketitle` with the various options we give it
  - ▶ `\title{}`
  - ▶ `\author{}`
  - ▶ `\date{}`
- ▶ A page of contents with `\tableofcontents`
- ▶ An abstract using `\begin{abstract}` environment
- ▶ Further customisation using `\begin{titlepage}` environment

# Formatting Documents: Sections

Hierarchy of  $\text{\LaTeX}$  is always important

1. `\section{}`
2. `\subsection{}`
3. `\subsubsection{}`

We can suppress numbering putting an asterix after the command, such as `\section*{}`

# Formatting Documents: Spacing

- ▶ Kerning exists
- ▶ `\,, \:, \;; \!` and `\` (with a space) give increasing spaces between words
- ▶ `\quad` gives space equal to current font size (from the quadrat!)
- ▶ `\vspace{<length>}` and `\hspace{<length>}`
- ▶ `\\` ends the line and starts another
- ▶ `\newline` will insert a new line altogether
- ▶ `\newpage`

Introduction

Basic Writing

Formatting Documents

**Mathematics**

Alignment and Arrays

Figures

Concluding Remarks

# Mathematics: Environments

- Instead of writing in *text* mode, we write in *math* mode

In-line	One-line	Multi-line
$\$ \$$		<del><math>\\$ \\$ \\$ \\$</math></del>
$\left( \right)$	$\left[ \right]$	
$\begin{math}$	$\begin{displaymath}$ $\begin{equation}$	$\begin{align}$

- $\left[ \right]$  will not number equations,  $\begin{equation}$  will
- We can suppress numbering using the  $*$

# Mathematics: Environments

## Inline and Oneline

An example of inline math to write  $x=2$ . Or an equation like:

```
\[ x=\frac{-b \pm \sqrt{ b^2-4ac }}{2a}\]
```

An example of inline math to write  $x = 2$ . Or an equation like:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# Mathematics: Environments

## Multiline

And a multi-line math environment such as

```
\begin{align*}
0 &= x^2 - 1 \\
0 &= (x-1)(x+1)
\end{align*}
```

And a multi-line math environment such as

$$\begin{aligned} 0 &= x^2 - 1 \\ 0 &= (x - 1)(x + 1) \end{aligned}$$

# Mathematics: Symbols

- ▶ Symbols are accessed via macros
- ▶ Greek symbols:
  - ▶ `\Omega` for uppercase  $\Omega$
  - ▶ `\omega` for lowercase  $\omega$
- ▶ Cheat sheet! You'll learn them eventually
- ▶ Some of my most common

$\int$	<code>\int</code>	$\implies$	<code>\implies</code>
$\partial$	<code>\partial</code>	$\cdot$	<code>\cdot</code>
$\pm$	<code>\pm</code>	$\leq$	<code>\leq</code>



# Mathematics: Superscripts and Subscripts

- ▶ we use `_` and `^`
- ▶ delimit its use with curly brackets `{ }` if more than one character is used

## Multiline

We can add subscripts like  $x_i$  or  $x_{i+1}$  \\  
Or superscripts like  $a^2$  and  $a^{n+1}$  \\  
Both can be used in conjunction like  $a_n^i$  \\  
But this can be improved using curly brackets as well;  $\{a_n\}^i$

We can add subscripts like  $x_i$  or  $x_{i+1}$   
Or superscripts like  $a^2$  and  $a^{n+1}$   
Both can be used in conjunction like  $a_n^i$   
But this can be improved using curly brackets as well;  $a_n^i$

# Mathematics: Operators

- ▶ `\sin`, `\cos`, `\tan` use `\sin`, `\cos`, `\tan` in math mode
- ▶ `\frac{}{}` creates a fraction
- ▶ `\dfrac{}{}` will force a *display style* fraction
- ▶ `\sqrt[]{}{}` will create a square root by default, or *nth* root if specified in `[]`
- ▶ `\mathop{dx}`  $dx$  for upright differential operators if you care
- ▶ `\text{}` to write text in math mode

## Various Operators

`$ \hat{a}, \dot{a}, \ddot{a}, \bar{a} $`

$\hat{a}, \dot{a}, \ddot{a}, \bar{a}$

# Mathematics: Delimiters

Along with the standard delimiters found on the keyboard,

$\langle$	<code>\langle</code>
$\rangle$	<code>\rangle</code>
$\{$	<code>\{</code>
$\}$	<code>\}</code>
$\ $	<code>\mid</code>

# Mathematics: Delimiters

- ▶ We can use *delimiters* to surround various expressions
- ▶ `\left<delimiter>` and `\right<delimiter>` to wrap using a delimiter
- ▶ Must have both commands to wrap
- ▶ use `\left.` if you do not want a left delimiter (or vice versa)

## Delimiters

```
\[ f(x,y)=\left| \frac{x+y}{2} \right| \]
```

$$f(x,y) = \left\| \frac{x+y}{2} \right\|$$



# Mathematics: Delimiters

## Delimiter Size Example

```
\[ \Big( 2\times \big( f(x,y)+g(x,y) \big) \Big)^2 \]
```

$$\left(2 \times (f(x,y) + g(x,y))\right)^2$$

Introduction

Basic Writing

Formatting Documents

Mathematics

Alignment and Arrays

Figures

Concluding Remarks

## Alignment and Arrays: Align

- ▶ The master key for alignment is the `&`
- ▶ In the `\begin{align}` environment, we can use it to align equations
- ▶ We separate lines using the `\\` macro

### Alignment

```
\begin{align*}
0 &= x^2+x\\
0 &= x(x+1)\\
&\implies x = 0, x=-1
\end{align*}
```

$$\begin{aligned} 0 &= x^2 + x \\ 0 &= x(x + 1) \\ \implies x &= 0, x = -1 \end{aligned}$$



# Alignment and Arrays: Arrays

- ▶ `\begin{array}{}` command
- ▶ takes alignment argument (c,l,r)
- ▶ each cell is separated by the `&`
- ▶ each line is separated by the `\\`
- ▶ we can use delimiters to form varying matrices!
- ▶ Array writes in math mode

# Alignment and Arrays: Arrays

## Example 2

```
\[  
\left(  
\begin{array}{cc}  
2-\lambda & 0\\4 & 3-\lambda\\ \end{array}  
\right)  
\]
```

$$\left( \begin{array}{cc} 2-\lambda & 0 \\ 4 & 3-\lambda \end{array} \right)$$

# Alignment and Arrays: Tables

- ▶ the text mode version is `\begin{tabular}{}{}`
- ▶ also takes an alignment argument
- ▶ `\hline` to make horizontal breaks
- ▶ `\usepackage{booktabs}` allows for fancy rulings
  - ▶ `\toprule`
  - ▶ `\midrule`
  - ▶ `\bottomrule`

# Alignment and Arrays: Tables

- ▶ `\usepackage{multicol}` and `\usepackage{multirow}` are super useful!
- ▶ `\multicolumn{n}{alignment}{text}`
  - ▶ combines n columns
  - ▶ the multicolumn has its own alignment options - need to but in borders if you'd like
- ▶ `\multirow{rows}{width}{text}` uses a similar format

Introduction

Basic Writing

Formatting Documents

Mathematics

Alignment and Arrays

**Figures**

Concluding Remarks

# Figures: Floats

- ▶ Floats cannot be broken across pages
- ▶  $\text{\LaTeX}$  will place floats *for* you where it makes sense
- ▶ Do not try to move floats to exactly where you want them -  $\text{\LaTeX}$  knows best!
- ▶ `\begin{tabular}` environments can be put into a `\begin{table}` to float

# Figures: Graphics

- ▶ we use `\usepackage{graphicx}` to import PNG, JPEG and PDFs
- ▶ `\includegraphics[parameters]{file path}`
  - ▶ `scale=`
  - ▶ `width=`
  - ▶ `height=`
  - ▶ `angle=`
- ▶ I would recommend putting your graphics into folders

# Figures

- ▶ We can specify figures using the `\begin{figure}[]` environment
- ▶ Everything inside our environment will be floated
- ▶ Figures can take placement parameters
  - h Try to place exactly here
  - t Try to place on top of page
  - b Try to place at bottom of page
  - p Place on separate page only for floats
- ▶ You can put as many as you want, and  $\text{\LaTeX}$  will go through them one by one



# Figures

- ▶ Commonly used with `\centering`
- ▶ `\caption{}`
- ▶ `\label{}` can be used to reference your figure

## Default Figure

```
\begin{figure}  
  \centering  
  \includegraphics{  
  \caption{Caption}  
  \label{fig:my_label}  
\end{figure}
```

# Figures: Referencing Figures

- ▶ We can use `\ref{label}` to call any figure or table
- ▶  $\text{\LaTeX}$  will count figures and tables separately
- ▶ The same can be done with equations if you label them (they will also be counted separately)

## Figures: Arranging Figures

- ▶ If you'd like to arrange figures in a specific way, you can use `\begin{minipage}{width}`
- ▶ `minipages` will align themselves next to each other if there is no whitespace
- ▶ perfect solution for arrays of figures



Figure: Frogs (2).

# Figures: Arranging Figures

The source code for the previous figure:

## Frogs Figure

```
\begin{figure}[h]
  \begin{minipage}{4cm}
    \includegraphics[width=4cm]{gra/frog.jpg}
  \end{minipage}
  \begin{minipage}{4cm}
    \includegraphics[width=4cm]{gra/frog.jpg}
  \end{minipage}
  \caption{Frogs (2).}
\end{figure}
```

Introduction

Basic Writing

Formatting Documents

Mathematics

Alignment and Arrays

Figures

Concluding Remarks

# Concluding Remarks: What Now?

- ▶ Practice!
- ▶ Explore new packages that may be of use to you
  - ▶ pgfplots
  - ▶ tikz
  - ▶ booktabs - lots more
  - ▶ mhchem - for chemical equations'
  - ▶ biblatex - referencing
- ▶ <https://tex.stackexchange.com/>

# Concluding Remarks: What Now?

- ▶ Tinker with other built in commands and formatting tools
- ▶ Make your own commands and macros
  - ▶ `\newcommand{}[]{}`
  - ▶ `\newcommand{\R}{\mathbb{R}}`
  - ▶ `\newcommand{\deriv}[1]{\frac{d}{d#1}}`

Thank You!