University of BRISTOL

Introduction To Scientific Computing

Basics of Latex

Dr. Fintan Healy
Room 1.31, Queens Building
fintan.healy@bristol.ac.uk

bristol.ac.uk

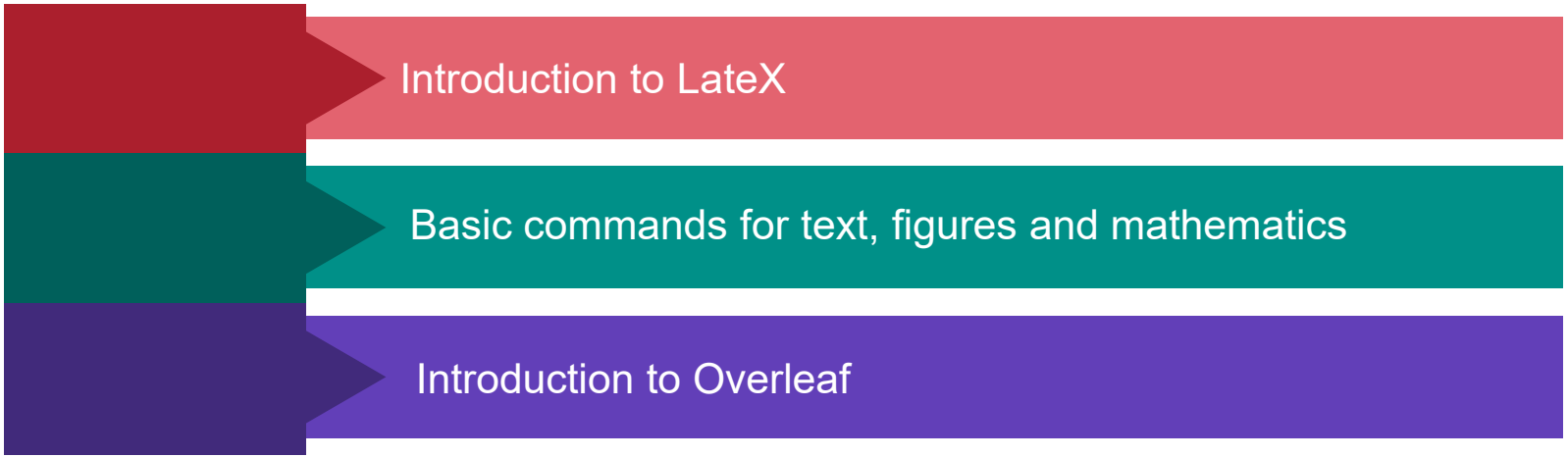
University of BRISTOL

Lecture 4

Basics of Latex

i2sc.fintanhealy.co.uk

Lecture Aims



University of BRISTOL

Academic Dissemination

bristol.ac.uk

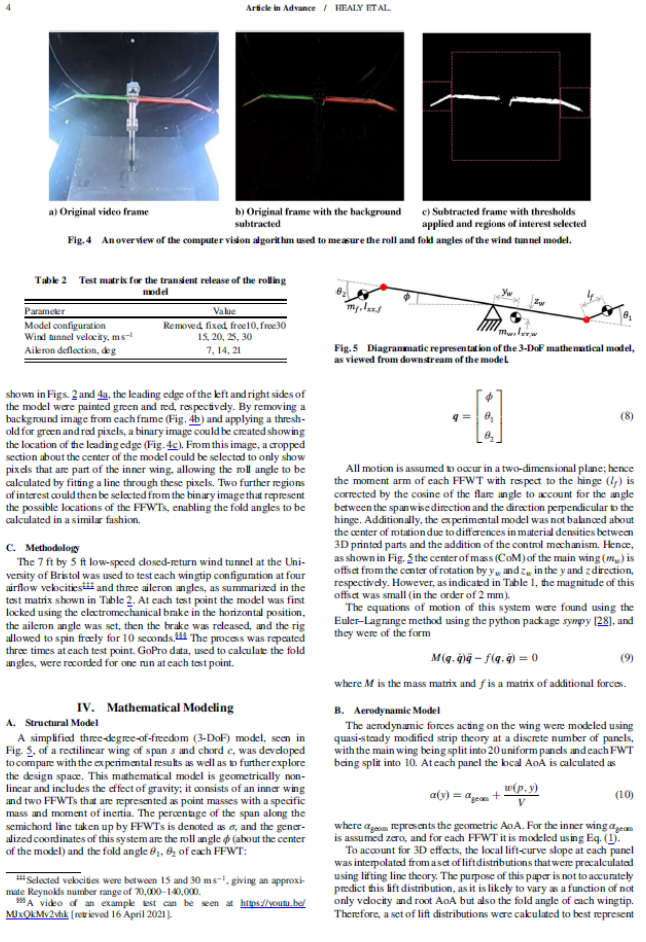
Academic Dissemination

- So far, we have learnt the basics of programming, including variables, loops, conditionals, functions, files, and arrays.
- And you've learnt programming in both Python and MATLAB
- But what's the best way to disseminate your results?

Academic Dissemination

The Challenge of Document Writing

- Technical and academic writing requires **precision, clarity, and consistency**.
- Common tasks include managing **equations, figures, references**, and formatting.
- Choosing the right tool impacts **productivity** and document **quality**.
  - Just like coding, you need to pick the right tool for the job



LaTeX versus Microsoft Word

Microsoft Word

- Drag and Drop user interface
- **WYSIWYG** (What you see is what you get) interface
- Familiar to most people (hopefully!)
- Equations, figures and citations possible but get tricky on large documents
- Large documents misbehave…
- Expensive*

LaTeX

- A “programming language”
- Allows you to **focus on the content, not the formatting**
- you write your document semantically e.g.
  - this is a section
  - this is a figure
  - this is a citation
- and LaTeX takes care of how it looks ensuring
  - Consistency
  - Elegance
  - typographic quality.
- Robust handling of figures, equations and citations etc…
- You must ‘compile the code’ to view the output
- Free*

Latex versus Microsoft Word

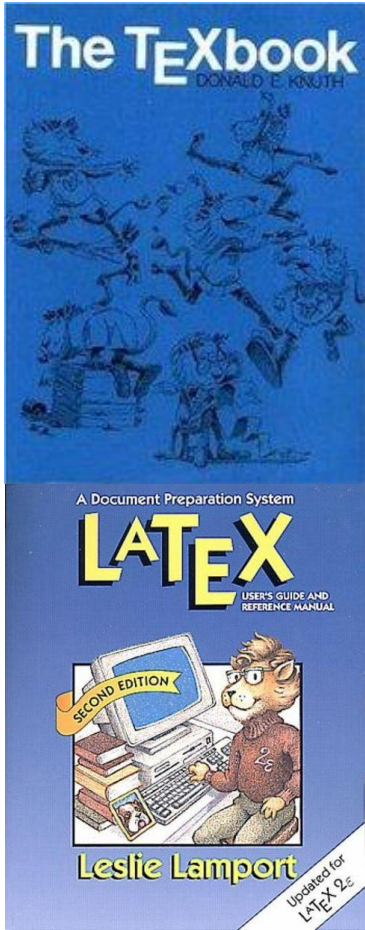
	Word	LaTeX
Ease of Use	Easy	Steep learning curve
Mathematic Support	Okay -> Good	Excellent
Collaboration	Good	Excellent (with overleaf)
Figures + Numbering	Okay	Excellent
Version Control	Poor	Good
Bibliography	Okay	Excellent
Formatting Control	Okay	Good

University of BRISTOL

LaTeX Basics

bristol.ac.uk

"tex", from the Greek word τέχνη (techne, meaning "art" or "craft").



LaTeX

- LaTeX is a markup language used to create scientific documents.
- Similar to writing and compiling a C program
  - A LaTeX document (".tex") is a text file with a series of commands
- As with other programming languages, there is a *reasonable* learning curve
- We will only scratch the surface today…
  - But there are lots of examples online
  - And LLMs are remarkably good at LaTeX

bristol.ac.uk

LaTeX (*.tex file)

```
\documentclass{article}

\usepackage{amsmath}
\usepackage{graphicx}

\title{My First Document}
\author{Fintan Healy}
\date{October 2025}

\begin{document}

\maketitle

\section{Introduction}

An example document to show you the basics of creating
documents in \LaTeX.

\subsection{Mathematics}

It is simple to create complex mathematical equations such as
\begin{equation}
\label{eq:myEq}
    y = \int^b_a \frac{1}{2} \sin(x) * (1 + \tan(x^2)) dx
\end{equation}

It is then easy to directly reference Eq.~\ref{eq:myEq} in the
text.

\end{document}
```

LaTeX Anatomy

LaTeX (*.tex file)

The type of document

```
\documentclass{article}
```

Load some packages

```
\usepackage{amsmath}
\usepackage{graphicx}
```

Preamble

```
\title{My First Document}
\author{Fintan Healy}
\date{October 2025}
```

Start the main body of the document

```
\begin{document}
```

Body

```
\maketitle

\section{Introduction}

An example document to show you the basics of creating
documents in \LaTeX.

\subsection{Mathematics}

It is simple to create complex mathematical equations such as
\begin{equation}
\label{eq:myEq}
    y = \int^b_a \frac{1}{2} \sin(x) * (1 + \tan(x^2)) dx
\end{equation}

It is then easy to directly reference Eq.~\ref{eq:myEq} in the
text.
```

Custom functions

Equations

Cross reference

End main body

```
\end{document}
```

LaTeX Anatomy

My First Document

Fintan Healy

October 2025

1 Introduction

An example document to show you the basics of creating documents in LaTeX.

1.1 Mathematics

It is simple to create complex mathematical equations such as

$$z = \int_a^b \frac{1}{2} \sin(x) * (1 + \tan(x^2))dx \tag{1}$$

It is then easy to directly reference Eq. 1 in the text.

LaTeX (*.tex file)

```
\documentclass{article}

\usepackage{amsmath}
\usepackage{graphicx}

\title{My First Document}
\author{Fintan Healy}
\date{October 2025}

\begin{document}

\maketitle

\section{Introduction}

An example document to show you the basics of creating
documents in \LaTeX.

\subsection{Mathematics}

It is simple to create complex mathematical equations such as
\begin{equation}
\label{eq:myEq}
    y = \int^b_a \frac{1}{2} \sin(x) * (1 + \tan(x^2)) dx
\end{equation}

It is then easy to directly reference Eq.~\ref{eq:myEq} in the
text.

\end{document}
```

LaTeX Anatomy

```latex
\documentclass{article}
```

- LATEX has different basic types of documents
  - article (most common)
  - letter
  - report
  - beamer (create slide decks)
  - Custom classes for journals etc…

```latex
\usepackage{}
```

- To use various functionalities, we need to load packages (like in Python)
  - Common packages are for loading images, entering mathematical formulae or citations

```latex
\begin{document}

% ----- the body -------

\end{document}
```

- Defines the start and end of the main body of the document.
  - Generally, all text should be written inside these comments.

Development Enviroments

- You can install LaTeX locally and compile documents offline
  - I recommend “TeX Live” coupled with “VS Code” (+ LaTeX Workshop extension)
- The easiest IDE is Overleaf. It is an online platform where:
  - Developing a Latex document is easy
  - Sharing documents is easy
  - Collaboratively working on documents is easy
- The final style of a LaTeX document can be customised
- Overleaf provides templates that you can use to get started.
  - https://www.overleaf.com/latex/templates

University of BRISTOL

Basic Syntax

bristol.ac.uk

Writing Text

- To write text:
  - separate words with spaces
  - separate sentences using periods
  - separate paragraphs by blank lines.
- When compiling, if multiple spaces are together, the compiler accepts these as just one continuous space.

LaTeX

```
We write a latex document as we would in any normal document .
Of course
the main difference is that the compiler takes care of the
formatting for         us.

Leave a blank line to start a new paragpraph.
But sentences.
Even short.
will be wrapped automatically, with   a blank line.

\noindent can be used to start a new paragraph without
indentation.
```

We write a latex document as we would in any normal document . Of course the main difference is that the compiler takes care of the formatting for us.

Leave a blank line to start a new paragraph. But sentences. Even short. Will be wrapped automatically, without a blank line.
can be used to start a new paragraph without indentation.

Text Formatting

- To **bold** and *italic* pieces of text we can

```
Use the “textbf” macro to \textbf{bolden text}, and the “textit” macro to \textit{italicise text}.
```

- The default font size in the article class is 10pt. The font size can also be changed from the default using:

```
\huge{This text} is huge. This is normal
sized, and this is \tiny{tiny}.
```

Command	Example text	Font size
\tiny	Some text	5pt
\scriptsize	Some text	7pt
\footnotesize	Some text	8pt
\small	Some text	9pt
\normalsize	Some text	10pt
\large	Some text	12pt
\Large	Some text	14.4pt
\LARGE	Some text	17.28pt
\huge	Some text	20.74pt
\Huge	Some text	24.88pt

Sectioning and Structure

- Scientific reports are usually structured in a logical numbering fashion, where there are sections and sub-sections within those sections:
- LaTeX automatically handles numbering. You just need to use
  - \section{}
  - \subsection{}
  - \subsubsection{}

LaTeX

```
\section{Introduction}
An example section.
\subsection{Aims}
With not one.
\subsection{Objectives}
But two sub sections. Some with their own sub sections
\subsubsection{Objective 1}
This is the first objective.
\subsubsection{Objective 2}
This is the second objective.
\section{Conclusion}
In conclusion, we have presented a document with sections and
subsections.
```

1 Introduction

An example section.

1.1 Aims

With not one.

1.2 Objectives

But two sub sections. Some with their own sub sections

1.2.1 Objective 1

This is the first objective.

1.2.2 Objective 2

This is the second objective.

2 Conclusion

In conclusion, we have presented a document with sections and subsections.

Lists

- Numbered lists or Bullet points are simple in LaTeX
  - \itemize – bullet points
  - \enumerate – numbered lists
- Each element starts with the keyword \item
- You can nest lists
  - You can mix and match

LaTeX

```
\begin{itemize}
    \item First element.
    \item Second element.
    \begin{itemize}
        \item A sub element.
        \item Another sub element.
    \end{itemize}
    \item Third element.
\end{itemize}
```

- First element.
- Second element.
  - A sub element.
  - Another sub element.
- Third element.

LaTeX

```
\begin{enumerate}
    \item First element.
    \item Second element.
    \begin{itemize}
        \item A sub element.
        \item Another sub element.
    \end{itemize}
    \item Third element.
\end{enumerate}
```

1. First element.
2. Second element.
   - A sub element.
   - Another sub element.
3. Third element.

Tables

- To create a table, we use the tabular environment.
- The speciation of the columns (i.e. justification) is then done using entries in a {} entry
  - l, c, or r for left, centre or right justification
  - | for vertical lines
- Use & to separate entries in a row
- Use \\ to create a new line
- Online tools such as tablesgenerator.com are great to help you quickly make tables

LaTeX

```
\begin{tabular}{|c|c c|}
    \hline
    Variable & Value & Units \\
    \hline
    $x$ & 5 & m \\
    $y$ & 10 & s \\
    \hline
\end{tabular}
```

Variable	Value	Units
x	5	m
y	10	s

Figures

- To load figures, we need to load the package graphic

```
\usepackage{graphicx}
```

- To include a figure
  - The file must be in the same directory
  - Use the command

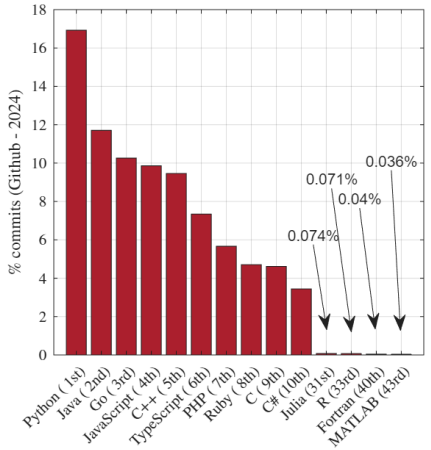
```
\includegraphics[width =0.5\linewidth]{LangChart.png}
```

- where '*langChart.png*' is the filename
- The optional argument 'width=0.5\linewidth' stretches the figure to half the total width of the page.
  - But if you've made your figures the right size, you don't need this!

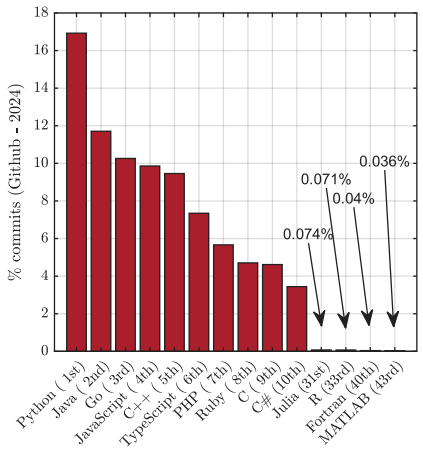
Raster Versus Vector Graphics

- When you save a figure, you can generally save it as either a raster or a vector graphic

Raster (*png*)



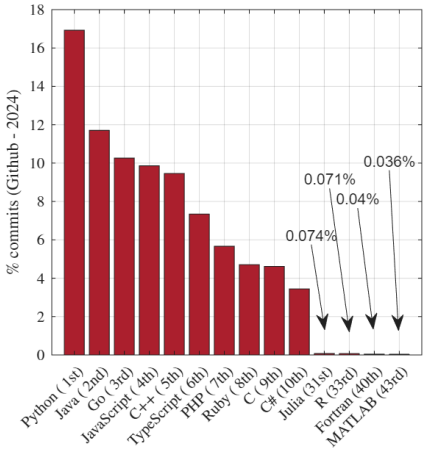
Vector (*pdf*)



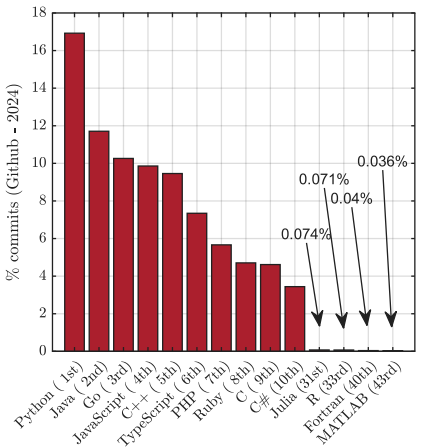
Raster Versus Vector Graphics

- When you save a figure, you can generally save it as either a raster or a vector graphic
  - **Raster graphics** are pixel-based images best suited for detailed visuals like photographs or complex surface plots
    - *.png .jpeg*
  - **Vector graphics** use mathematical paths to create scalable and resolution-independent illustrations
    - *.svg .pdf*
- If your report will be read digitally, vector graphics give it a more professional feel…
  - However, they don't always play nice…

Raster (*png*)

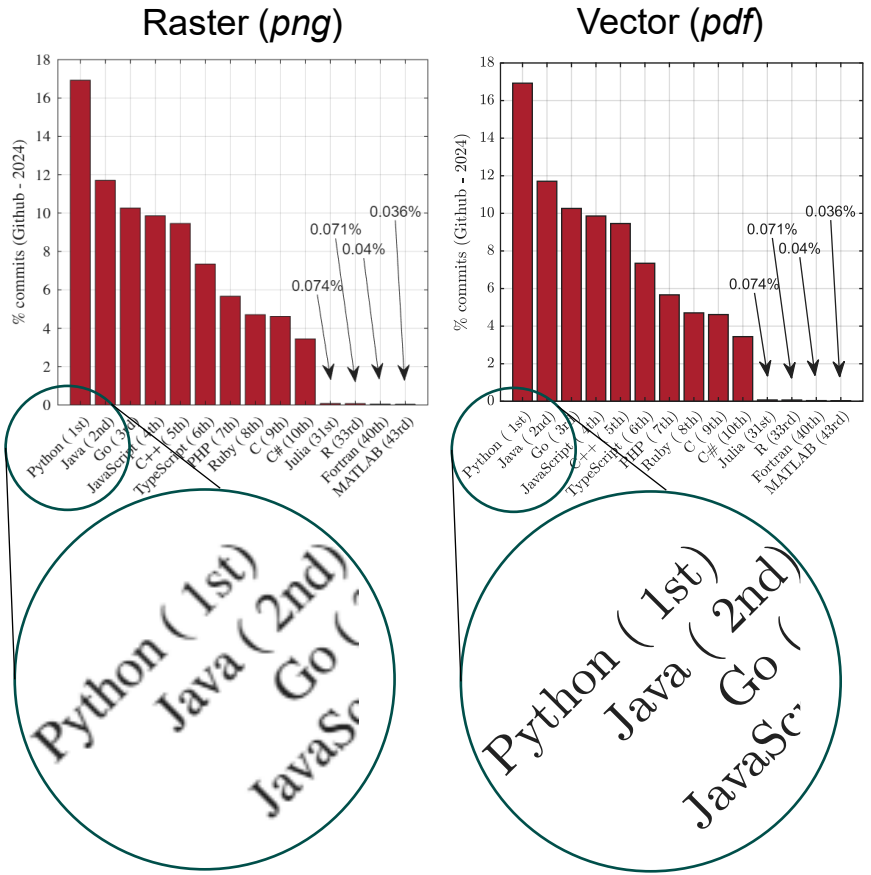


Vector (*pdf*)



Raster Versus Vector Graphics

- When you save a figure, you can generally save it as either a raster or a vector graphic
  - **Raster graphics** are pixel-based images best suited for detailed visuals like photographs or complex surface plots
    - *.png .jpeg*
  - **Vector graphics** use mathematical paths to create scalable and resolution-independent illustrations
    - *.svg .pdf*
- If your report will be read digitally, vector graphics give it a more professional feel…
  - However, they don't always play nice…



Floats, Captions and Placement

- We can encapsulate figures and tables within a floating environment
- This lets LaTeX decide where to place the element
  - You "***focus on the content, not the formatting***"…
- For figures, we can use a figure environment
  - Use \caption to add a … caption
  - Use \label to be able to refer to the figure in the text
- The optional placement specifier (e.g., [h!]) suggests preferred float positioning
  - E.g., h means 'here' and ! means "try harder". There is also t – top, b – bottom, etc.…
- You can constrain the movement of floats with the macro \FloatBarrier

```
\usepackage{placeins}
\FloatBarrier
```

LaTeX

```
\begin{figure}
    \centering
    \includegraphics[width =0.5\linewidth]{LangChart.png}
    \caption{A simple figure with a caption.}
    \label{fig:lang}
\end{figure}

A reference to Fig.~\ref{fig:lang} in the text.
```

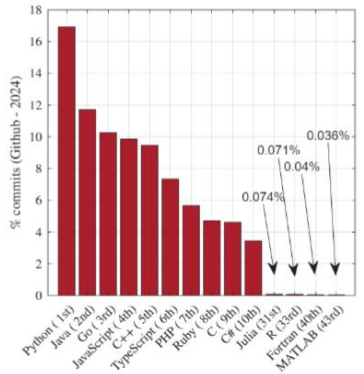


Figure 1: A simple figure with a caption.

A reference to Fig. 1 in the text.

Floats, Captions and Placement

- Tables can be encapsulated in a ‘tabular’ environment
- By convention, captions should be shown
  - Before tables
  - After figures
- \centering centre justifies content in the ‘environment’

LaTeX

```
\begin{table}
    \centering
    \caption{A simple table with three columns.}
    \label{tab:simple}
    \begin{tabular}{|c|c c|}
        \hline
        Variable & Value & Units \\
        \hline
        $x$ & 5 & m \\
        $y$ & 10 & s \\
        \hline
    \end{tabular}
\end{table}
```

Table 1: A simple table with three columns.

Variable	Value	Units
x	5	m
y	10	s

Sub-figures

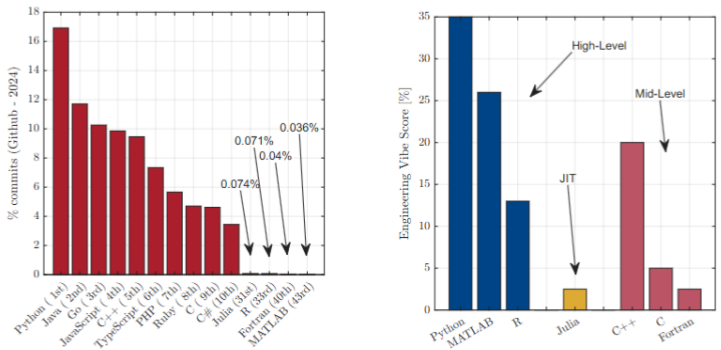
- In many circumstances, we want a figure to have multiple sub-figures.
- To create subfigures, each with its own caption, load the package

```
\usepackage{subcaption}
```

- Then have multiple subfigure environments in one figure environment
- The size of each figure drives the layout

LaTeX

```
\begin{figure}
    \centering
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{LangChart.pdf}
        \caption{A subfigure.}
        \label{fig:lang_sub}
    \end{subfigure}
    \hfill
    \begin{subfigure}{0.45\linewidth}
        \includegraphics[width=\linewidth]{LangVibe.pdf}
        \caption{Another subfigure.}
        \label{fig:lang_sub2}
    \end{subfigure}
    \caption{A figure with subfigures.}
    \label{fig:lang_main}
\end{figure}
```



(a) A subfigure. (b) Another subfigure.

Figure 2: A figure with subfigures.

Maths

- One of the great strengths of LaTeX is equation writing
- We will only cover the basics here, but more details can be found in the references.
- It's best to load the package amsmath
- Equations are written in a maths environment, use:
  - $ … $ for inline mathematics
  - \begin{equation} ... \end{eqaution} for number equations
- You can also label and cross-reference equations

LaTeX

```
Simple equations in the text: $y = mx + c$ or in a separate line
\begin{equation}
    L = \dfrac{1}{2} \rho v^2 S C_L
    \label{eq:lift}
\end{equation}
which is known as the lift equation (Eq.~(\ref{eq:lift})).
```

Simple equations in the text: $y = mx + c$ or in a separate line

$$L = \frac{1}{2}\rho v^2 S C_L \tag{1}$$

which is known as the lift equation (Eq. (1)).

Maths

Feature	LaTeX	Output
Sub- or superscripts	`f(x)=a_{0}+ a_{1} x+a_{2} x^{2}+\dots`	$f(x) = a_0 + a_1x + a_2x^2 + \dots$
Greek Letters	`\alpha , \beta , \gamma , \Gamma , \pi , \Pi , \phi`	$\alpha, \beta, \gamma, \Gamma, \pi, \Pi, \phi$
Bold font (e.g for vectors)	`\mathbf{a} \cdot \mathbf{b} = | \mathbf{a} || \mathbf{b} | \cos\theta`	$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|\|\mathbf{b}\| \cos\theta$
Fractions	`R = \frac{\rho V L}{\mu}`	$R = \frac{\rho V L}{\mu}$
Large Brackets “\left(… \right)”	`\left(\frac{V_{1}}{V_{2}}\right)^{\gamma}`	$\left(\frac{V_1}{V_2}\right)^{\gamma}$
Summation and integration	`\int_{a}^{b} f(x) dx \approx \sum_{i=1}^{N} \Delta x f(x_i)`	$\int_a^b f(x)dx \approx \sum_{i=1}^{N} \Delta x f(x_i)$

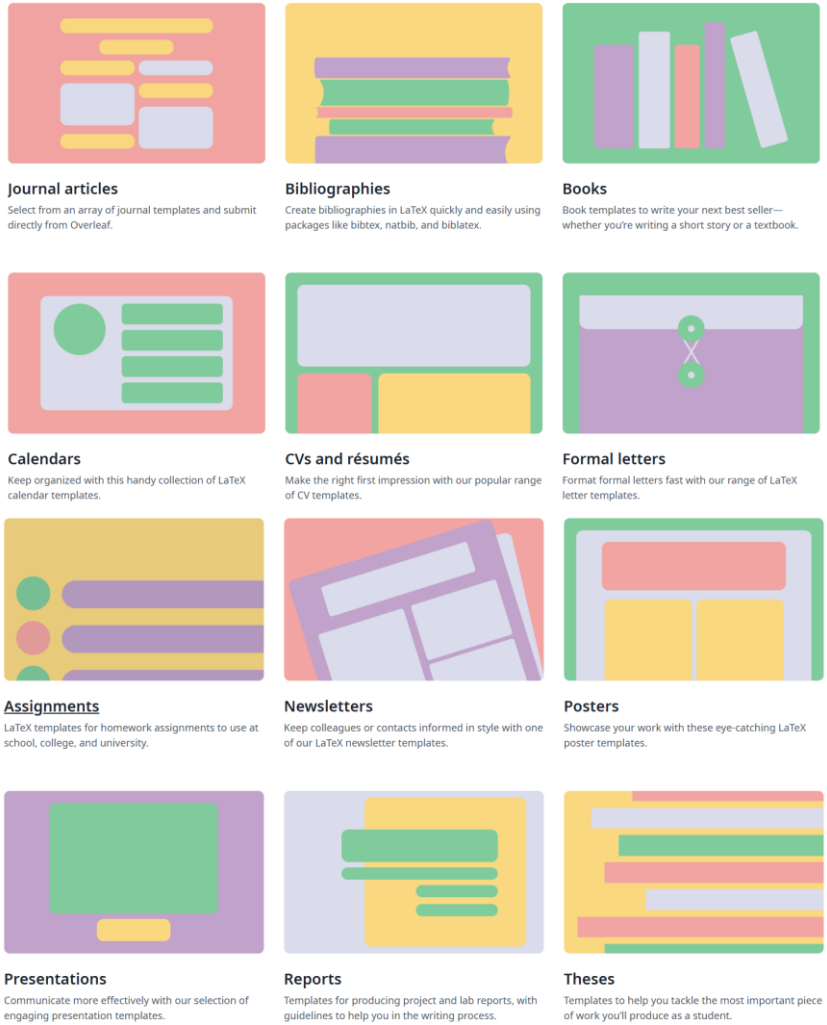
More Documentation

- We have only scratched the surface of LaTeX
- For further details, please start with Overleaf's 30-minute guide https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes
- And the overleaf documentation in general is excellent
- LaTeX also has an online wiki https://en.wikibooks.org/wiki/LaTeX
- And LLMs are generally very good at LaTeX

Templates

- Don’t forget to start with a template you like the look of
- These *control* the general styling of your document.

https://www.overleaf.com/latex/templates



University of BRISTOL

SUMMARY

bristol.ac.uk

Summary

- Have introduced software for producing professional scientific reports (LaTeX)
- Looked at:
  - Basic formatting
  - Import figures and tables
  - Writing equations

https://i2sc.fintanhealy.co.uk/

Moving Forward

- **No more labs – but your journey is not over!**
  - Attempt the rest of the workbooks; they will save you time when it comes to your coursework!
- I will release solutions at the start of week 5
- **Don't neglect Python**
  - Being 'conversational' in multiple programming languages is a good skill for the 21st century
- Have a go creating a report with LaTeX