Thesis Title

by

Author Name

A Thesis
presented to
University Name

In partial fulfilment of requirements
for the degree of
Degree Name
in
Program Name

City, Country

ABSTRACT

THESIS TITLE

Author Name
University Name, 2018

Advisor:
Prof. Advisor

LaTeX is beautiful. Everybody should use it. Unfortunately, it's a little hard to learn. But you only have to do it once.

Acknowledgments

Thanks Mom.

"You gonna learn today"

— Kevin Hart

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we introduce LaTeX.

1.1 Comparison vs. Microsoft Word

LaTeX is an alternative to MS Word. It's not better in every situation (Figure 1.1), but for large, complicated, or math-heavy documents, its undeniably better.

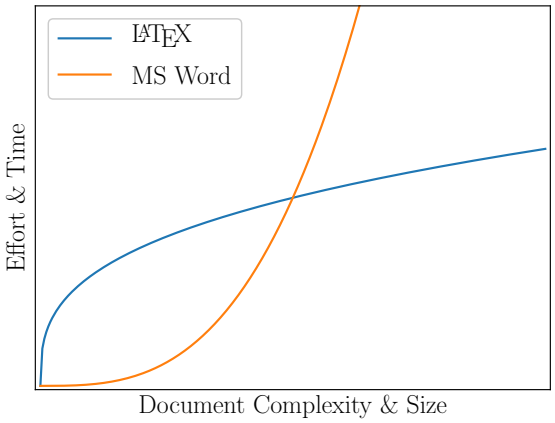

Figure 1.1: LaTeX vs MS Word

The major paradigm shift when transitioning from MS Word to LaTeX is that you do not see the final product while you are typing. Instead, the document is *compiled*, producing a PDF document directly.

This allows more complex document creation, without trying to resolve all links as you type. For example,

the Table of Contents for this document is generated automatically using `\tableofcontents`.

1.2 Components

1.2.1 Commands

In LaTeX, all commands are preceded by a backslash: `\`. Not all commands accept arguments, but when they do, you must supply all of them, and enclose each in braces: `{...}`. For example, `$\frac{1}{2}$` produces $\frac{1}{2}$ (more on the `$` later).

You can create your own commands using: `\newcommand{\cmd}{definition}`, which you could use as: `my command is: \cmd`, producing: "my command is: definition". If you want your command to use arguments, the syntax is: `\newcommand{\cmd}[1]{You provided the argument: #1}`, for which `\cmd{apples}` produces "You provided the argument: apples".

1.2.2 Packages

LaTeX doesn't always have all the tools we need. We'll often need to import user-created packages to do some fancy stuff, like embedding images, creating flow-charts, etc. To do this, we use: `\usepackage{packagename}`. This will provide us additional commands that we can use to create our document. You must load packages in the document header.

1.2.3 Hello World

What is the document header, you ask? Let's look at a minimal working example. This code produces a document containing only the words "hello world!":

```
\documentclass{article}
% load packages here
\begin{document}
  % document content here
  hello world!
\end{document}
```

The `\documentclass` command specifies the type of document, while the rendered contents of the document are enclosed in `\begin{document} ... \end{document}`. Comments can be added using `%`.

Chapter 2

Math

LaTeX is probably most well known for typesetting beautiful math.

You can write inline variables like: `$\pi$` → π or small equations like: `$\pi = 3.14$` → $\pi = 3.14$.

Simple stand-alone equations can be invoked several ways, including `\[...\]`, `$$...$$`:

$$x = 1$$

$$y = 2$$

but these are not numbered in the margins, and we can't reference them elsewhere.

To number the equation, we need to use an environment, like: `\begin{equation} ... \end{equation}`:

$$z = 3 \tag{2.1}$$

If we want to reference the equation, we also need to place a `\label{keyword}` within the environment. Then, elsewhere we can use `Eq. (\ref{keyword})` to reference like this: Eq. (2.1).[1] It's best practice to use something like `eq:z` for the `keyword` in equations, `fig:x-vs-t` for figures, etc.

For more complicated or equations, the `amsmath` package provides additional helpful environments, like

[1] Here we're also using the `hyperref` package with the `colorlinks` option, which colours the reference and makes it clickable. Same thing for footnotes like this.

`alignat`:

$$\frac{dS}{dt} = -(\lambda + \mu)S + (\gamma \qquad)I + (\mu)N \tag{2.2}$$

$$\frac{dI}{dt} = +(\lambda \qquad)S - (\gamma + \mu)I \tag{2.3}$$

though we might want to re-write that using some "`array`"s:

$$\frac{d}{dt}\begin{bmatrix} S \\ I \end{bmatrix} = \begin{bmatrix} -\lambda - \mu & +\gamma \\ +\lambda & -\gamma - \mu \end{bmatrix}\begin{bmatrix} S \\ I \end{bmatrix} + \begin{bmatrix} \mu \\ 0 \end{bmatrix} N \tag{2.4}$$

You get the idea.

Chapter 3

Figures, Tables, & Citations

In this chapter, we'll discuss how to add Figures, Tables, and Citations to a LaTeX document.

3.1 Figures

In LaTeX, Figures and images are not the same thing. Figures are an environment, while images represent their content (mostly). This is actually very powerful, since it allows you to combine multiple images, text, flow-charts, or whatever you like in a single Figure.

The Figure environment is invoked like equations: `\begin{figure} ... \end{figure}`.

To embed images, we need to use the `graphicx` package, and also tell it where to look for images using `\graphicspath{{folder/},{anotherfolder/},...}`. Then, we can use `\includegraphics{imagename}` to add the image to the `figure` environment.

To caption the figure, we use the `\caption{caption text}` command within the `figure` environment, which automatically numbers the Figure based on the document class.

Just like equations, we can use `\label{}` and `\ref{}` to reference the Figure number elsewhere. The nice part is, whenever Figures are moved around, LaTeX takes care of the number ordering, and the in-text references always point to the same Figure.

Putting it all together:

Code:

```
Figure \ref{fig:devs} shows Ballmer shortly after the peak.
\begin{figure}[H]
  \centering\includegraphics[width=0.5\textwidth]{developers.jpg}
  \caption{Developers}\label{fig:devs}
\end{figure}
```

Result:

Figure 3.1 shows Ballmer shortly after the peak.

Figure 3.1: Developers

3.1.1 Extra Tips

Its usually good to use `\centering` before `\includegraphics` to centre the image.

The command `\includegraphics` also accepts optional arguments for `width=...`, `height=...`, etc. Best practice is to set this to `0.5\textwidth`, or some proportion other than 0.5, so that the image adapts to the margins. Note that `0.5*\textwidth` will give an error – the multiplication of lengths is implicit.

3.1.2 Tikz

The `tikz` package provides tools for creating flow-charts. The flow-chart is written using LaTeX code, but best practice is to write the code in a separate file, and then include the file using `\include{filename}`.

Figure 3.2 gives an example flow-chart (namely, the contents of this document). The syntax of `tikz` is essentially a language all of its own, and we're not going to delve into it any further here. In B.1 Tikz

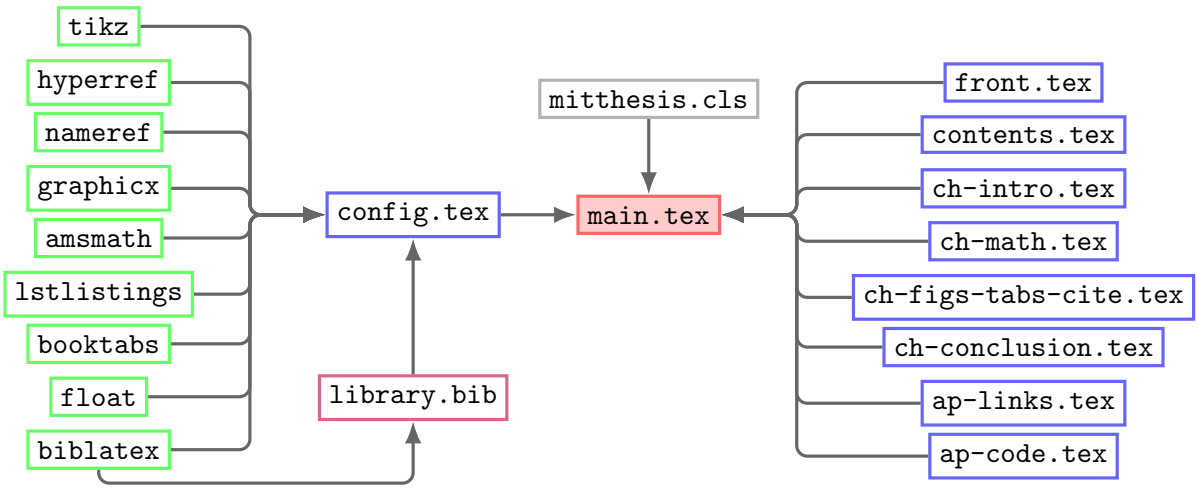

Figure 3.2: Overview of this document

Code, you can see the code used to create Figure 3.2.

3.2 Tables

Tables are almost identical to Figures, except this time the content is a `tabular` environment, and the table object is a `table` environment.

You can specify the alignment of each column using the argument to a `tabular` environment: `\begin{tabular}{lcr} ... \end{tabular}`, where `lcr` defines 3 columns, having **l**eft, **c**enter, and **r**ight alignment. Within the table content, rows are separated by newlines: `\\`, while columns are separated by the `&` symbol. Here is an example:

Code:

```
\begin{table}[H]
  \centering\caption{Windows Operating Systems}
  \vspace{0.5em}
  \begin{tabular}{ccl}\toprule
        Year & OS & Reviews     \\ \midrule
        2009 & 7  & a triumph   \\
        2012 & 8  & a disaster  \\
        2016 & 10 & meh         \\ \bottomrule
  \end{tabular}
\end{table}
```

Result:

Table 3.1: Windows Operating Systems

Year	OS	Reviews
2009	7	a triumph
2012	8	a disaster
2016	10	meh

3.3 Citations

Automatically generating citations is another helpful feature of LaTeX. However, setting up the workflow can be a little tricky.

First, you'll need a bibliography file. These can be exported from reference managers like Mendeley or Zotero. In B.2 Example Bibliography File an example is given.

Code:

Within your document header, you need to load the package `biblatex` (with whichever formatting options suit you), and specify your bibliography file: `\bibliography{library.bib}`.

Then, within the body of the document, you can use the `\cite{...}` command to cite a reference, replacing `...` with the `id` of the desired entry in your `.bib` file (e.g. `Ashburner2005`).

Finally, wherever you want the bibliography to appear, you can use the `\printbibliography` command.

Compiling:

We're not out of the woods yet. The document compiler, `latex`, doesn't actually do reference management. There are several other tools which can do this (and actually, choosing one gets complicated). A good choice here however, is `bibtex`.

What you need to do is: compile your document with `latex`, then again with `bibtex`, and then one last time with `latex` again. For a document called `main.tex`, this looks like:

```
$ latex main
$ bibtex main
$ latex main
```

What happens is: `latex` compiles the document as much as possible, putting placeholders for every citation, and producing a list of the missing references (and formatting specifications) in an auxiliary file (`main.aux`); `bibtex` then looks for `main.aux`, and uses the specifications and bibliography file to create one more file, `main.bbl`, which provides `latex` with formatted answers for each of the citation placeholders; the second pass of `latex` then finds `main.bbl` and replaces the placeholders (including the bibliography itself) with the formatted citation information.

3.3.1 Example

Code:

```
Unified Segmentation by \citeauthor{Ashburner2005} (\citeyear{Ashburner2005})
is a popular model for segmenting brain MRI \cite{Ashburner2005}.
\subsubsection*{References}
\printbibliography[heading=none]
```

Result:

Unified Segmentation by **Ashburner2005** (**Ashburner2005**) is a popular model for segmenting brain MRI [**Ashburner2005**].

References

Chapter 4

Conclusion

That's it! Be sure to check out Appendix A Useful Links for some additional resources.

Happy LaTeX-ing.

Appendix A

Useful Links

LaTeX Install Guide – You'll need to do this first

TeX Stack Exchange – Q & A style how-to and debugging help

TeXstudio – The best LaTeX writing desktop application

Texmaker – Another LaTeX writing desktop application

Overleaf – An online LaTeX writing online application

LaTeX Cheat Sheet – A really nice reference for common commands

Google – You're going to need it :)

Appendix B

Code

B.1 Tikz Code

```
\tikzset{%
  arrow/.style    = { ->,
                      very thick,
                      rounded corners,
                      >     = Latex,
                      draw  = black!60!white },
  element/.style  = { very thick,
                      fill  = white,
                      draw  = #1!60!white,
                      align = center,
                      font  = \ttfamily },
}
\begin{tikzpicture}
  \node(main)  [element=red,fill=red!20!white]{main.tex};
  \node(config)[element=blue,left = 1cm of main]{config.tex};
  \node(cls)   [element=gray,above = 1cm of main]{mitthesis.cls};
  \draw[arrow] (config) -- (main);
  \draw[arrow] (cls) -- (main);

  \coordinate [above right = 1.75cm and 4cm of main] (files);
  \coordinate [right = 1cm of main] (fx);
  \foreach \i/\file in {
     0/front,
     1/contents,
     2/ch-intro,
     3/ch-math,
     4/ch-figs-tabs-cite,
     5/ch-conclusion,
     6/ap-links,
     7/ap-code%
   }{
      \node(\file) [element=blue,below = 0.7*\i cm of files]{\file.tex};
      \draw[arrow] (\file) -| (fx) -- (main);
  }
  \coordinate [above left = 2.45cm and 3cm of config] (files);
  \coordinate [left = 1cm of config] (fx);
  \coordinate [below = 3.55cm of fx] (xl);
```

```
  \foreach \i/\file in {
    0/tikz,
    1/hyperref,
    2/nameref,
    3/graphicx,
    4/amsmath,
    5/lstlistings,
    6/booktabs,
    7/float,
    8/biblatex%
  }{
      \node(\file) [element=green,below = 0.7*\i cm of files]{\file};
      \draw[arrow] (\file) -| (fx) -- (config);
  }
  \node(lib)[element=purple, below = 1.8cm of config] {library.bib};
  \draw[arrow] (lib) -- (config);
  \draw[arrow] (biblatex.south) |- (xl) -| (lib.south);
\end{tikzpicture}
```

B.2 Example Bibliography File

```
@article{Ashburner2005,
  abstract = {A probabilistic framework is presented that enables
  image registration, tissue classification, and bias correction
  to be combined within the same generative model.
  A derivation of a log-likelihood objective function
  for the unified model is provided.
  The model is based on a mixture of Gaussians and is extended to incorporate
  a smooth intensity variation and nonlinear registration with tissue probability maps.
  A strategy for optimising the model parameters is described,
  along with the requisite partial derivatives of the objective function.},
  author = {Ashburner, John and Friston, Karl J},
  doi = {10.1016/j.neuroimage.2005.02.018},
  issn = {1053-8119},
  journal = {NeuroImage},
  keywords = {Algorithms,Brain Mapping,Computer-Assisted,Data Interpretation,
  Fuzzy Logic,Image Processing,Likelihood Functions,Magnetic Resonance Imaging,
  Models,Neurological,Nonlinear Dynamics,Normal Distribution,Probability Theory,
  Statistical,statistics {\&} numerical data},
  number = {3},
  pages = {839--851},
  pmid = {15955494},
  title = {{Unified segmentation.}},
  volume = {26},
  year = {2005}
}
```