Oxford Thesis Template Example

FIRST LAST

XXX College

University of Oxford

A thesis submitted for the degree of

Something or Other

Trinity term, 2025

Supervised by Supervisor A and Supervisor B

Abstract

Welcome to an updated LaTeX template for Oxford University theses! This is not officially sanctioned by the University of Oxford whatsoever, merely the combined efforts of students and faculty members over the past couple of decades.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Acknowledgements

Thanks must go to various OU members who have come before me, namely Keith Gillow, who created the OCIAM thesis template, and then Sam Evans, John McManigle, and Robin Scales, who have built up this template since.

Contents

Contents

Abbreviations

OCIAM Oxford Centre for Industrial and Applied Mathematics

OU   Oxford University

Glossary

LaTeX Is a markup language specially suited for scientific documents

1

Introduction

1.1 A section

Here is an example of how a chapter page looks, without a minitoc (aka **mini table of contents**).
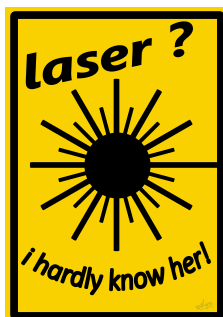
This is a section.

1.1.1 A subsection

This is a subsection.

Corrections can be inserted as so.

Shorter corrections can be inserted like so.

Figure 1.1: This is a figure. Courtesy of Suzanne Lim (CC BY-NC-ND 4.0).

1.2 Why LaTeX?

1.2.1 Working At A Chapter At A Time

A **major** benefit of LaTeX over Word is that you can write chapters, sections, etc as separate files, and then "include" or "input" them into the document being compiled (\include{} & \input{}).

Unlike Word, you need to compile your document, which sounds daunting but as simple as the big green "Recompile" button or "ctr+s" on your keyboard.

The main file of this document is Oxford_Thesis.tex. Which has \include{} commands in each of the chapters being displayed right now. If one wanted to not show a chapter in this final document, then all you have to do is comment it out with "%". You can use \input{} within chapters to insert section or subsection files etc.

Now you do not have to delete sections or chapters that you have previously written; rearranging chapters or sections is as simple as rearranging lines of code; you can compile the whole thesis as one final document less to worry with cross-referencing and citations.

1.2.2 Citations & Reference Managers

Citations in LaTeX do not slow down nearly as much as in Word in terms of adding and updating them. They are relatively easy to add with the \cite{} command which quickly searches your reference file (in this case it's references.bib), e.g., \cite{whitehead1927principia} produces [1]. **Additionally**, with premium Overleaf, you can synch your Zotero, Mendeley, etc reference manager to your document, making it very easy. **Otherwise** you would need to export out and update your reference file manually.

1.2.3 Packages & Commands

The LaTeX community is vast, and so, many clever people have worked to develop packages that you can easily add and use into your own work, adding extra

functionality and quality-of-life features. I (Robin) have written an extra chapter (chapter 2) just to highlight and give examples and usages of all of the useful packages and their commands that I used in my own thesis.

It will additionally help you to understand how to write LaTeX too by looking at generated content and relating that to the source-code `ch-02-Examples.tex` file.

2

How To Use This Template & LaTeX

Contents

Firstly, many thanks to Q for making my (Robin) version of the template more concise.

All of LaTeX's useful packages that I and others have found are located with

`OxfordTexThesis.cls` within the "RJS Packages" block of lines. Below will outline some examples of what they are and how they can be very useful. What I will do is have the outputted of the code shown followed by what was written in the LATEX script using (`this`) format.

If you are on Overleaf with this template, you can double-click on any of the features in this chapter, and it should bring you to the `ch-02-Examples.tex` file and the location where you double-clicked. This is a way to learn how to use LATEX alongside lots of good tutorials online.

2.1 Drafting

Q has nicely added in a clear "Draft" watermark over the document to make it clear that the document is a draft. However, it may interact with pdf editors when you or your supervisor wants to highlight text. Additionally, it can be interacted with in Overleaf when double-clicked. To turn it off, see the following in `Oxford_Thesis.tex` file.

```
% To include a "Draft" watermark, comment below.
% \backgroundsetup{contents={}}
```

John McManigle's version of the Thesis Template, which you can find on GitHub, has a simple draft and date at the bottom of the page but is then less obvious.

2.2 Figures

2.2.1 subcaption

The `subcaption` package allows you to have one figure, like fig. 2.1, with multiple subfigures. Each one can have its own caption and its own label so that you can cross-reference a specific one. So you can reference the bitmap figure (fig. 2.1a) and vector figure (fig. 2.1b) separately from each other and the main figure. One of the **major benefits** of it is that rather than creating complex figures in PowerPoint or Inkscape, you can make them within LATEX and quickly change elements within it.

(a) Bitmap   **(b)** Vector
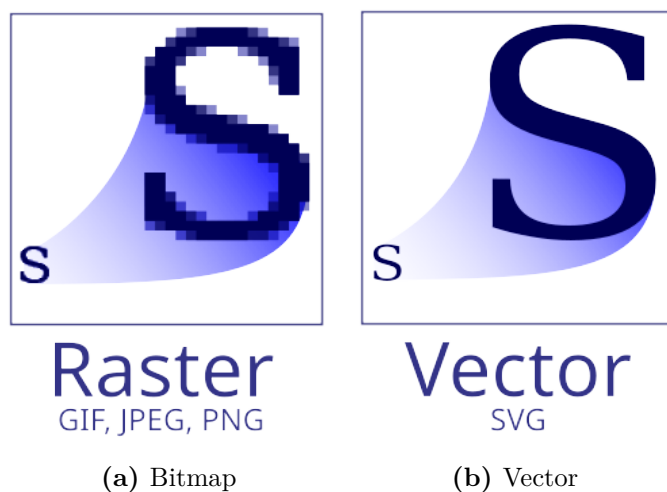
Figure 2.1: Example of subfigure package. Yug, modifications by Cfaerber et al., CC BY-SA 2.5 <https://creativecommons.org/licenses/by-sa/2.5>, via Wikimedia Commons.

2.2.2   inkscapelatex

As hinted in fig. 2.1, vector graphics can be very useful when it comes to plots, diagrams, or anything that is not a photograph/micrograph, so that the image remains "crisp" no matter the displayed view. If you increase the zoom in fig. 2.2 it will always remain crisp as it is an "svg" image file. In contrast, the images in fig. 2.1 are "png" files and so are raster type images, so increasing the magnification it will begin to pixelate.

MATLAB, Python, and I believe PowerPoint can all save images as *svg* files which can then be directly used in L#T#X using `\includesvg[]{}`.

Importantly, svgs are the native file format for Inkscape (download here) which is a vector image editing software. I think this is more powerful for the following reasons:

1. Free software used by many people with loads of tutorials and forum posts online.

2. You can open raster images (e.g. micrographs) and add in annotations with layers over it, then export them as pngs are jpgs. So, you can look through folders of images in a thesis, see the preview for that figure, go back in, edit the annotation, and make quick changes for your supervisor.
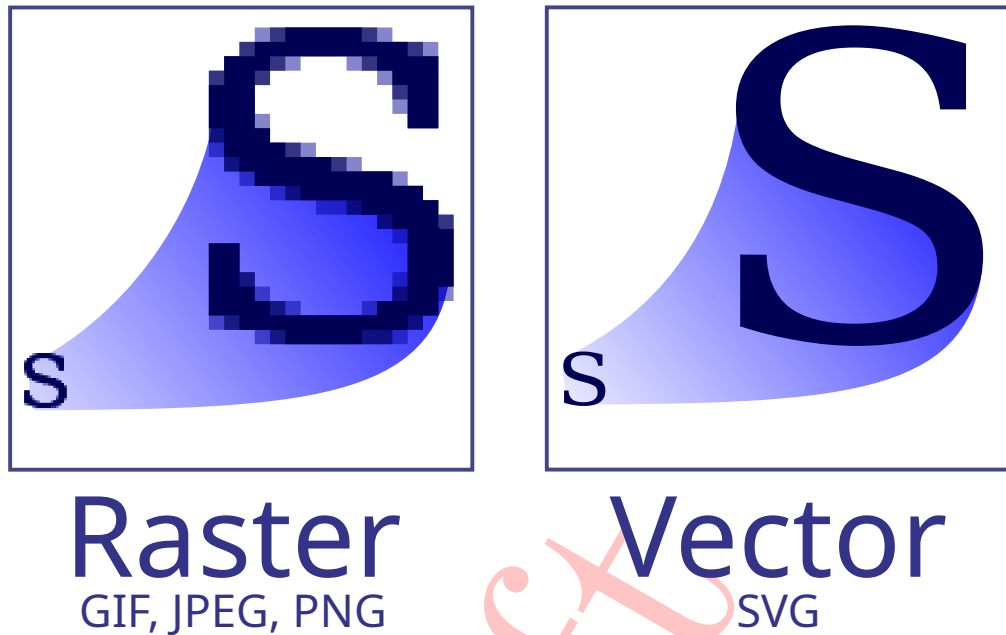
Figure 2.2: Example of scalable vector graphics (*svg*) image files being displayed in LATEX using the `inkscapelatex` package. Yug, modifications by Cfaerber et al., CC BY-SA 2.5 <https://creativecommons.org/licenses/by-sa/2.5>, via Wikimedia Commons.

3. Sometimes you can edit the text in saved MATLAB or Python SVG figure files. Or make another changes like removing the background if you want to do that later.

4. The layers and organisation of Inkscape is really good in comparison to PowerPoint.

5. Making diagrams is the same or easier as it is PowerPoint.

2.2.3 pdflscape & makecell

pdflscape is useful for when you sometimes need a horizontal page for especially wide figures or tables.

makecell is great for adding in multiple rows into the same cell. This forum post is useful for formatting it.

Table 2.1: Example table with the formatting which I like.

Left Aligned Column 1	Central Aligned Column 2	Right Aligned Column 3	Column 4	Column 5	Column 6	Col 7
Reference [1]	1	g	makecell is useful for multiple rows			
b	2	h	7			
c	3	i	8			
d	4	j	9			
e	5	k	10			
f	6	l	$y = x^2$			

2.2.4 pgf-pie

`pgf-pie` allows you to generate pie charts within the document itself.

2.2.5 placeins

`placeins` was useful for me to enforce spaces between floats (e.g. figures) and text. LATEX tries to be clever and maximise the efficiency of the text. However, that sometimes means that the text and its related figure are too far apart, or another section begins awkwardly. Hence, the `\FloatBarrier` command stops text from occurring before that float where that barrier is placed. This Reddit post is where I got it from initially.

Similarly, `\clearpage` as mentioned in that post creates a page break and is even used in this chapter.

2.3 Text

2.3.1 Acronyms

This subsection is in development! This section will explain how to use acronyms so that you always make sure that you have already defined an acronym. It makes sure that the first instance in the document is written in full and then abbreviated later in the document.

2.3.2 cleverref

`cleverref` is essentially clever hyperlink referencing. In LATEX the command `\ref{}` only puts in the ID of the label it is cross-referencing. For example, using the above figure with `\label{fig: svg example}`

- Default `\ref{fig: svg example}` command → "2.2"
- Cleverref `\cref{fig: svg example}` → "fig. 2.2"
- Cleverref `\Cref{fig: svg example}` → "Figure 2.2"

Hence, it automatically identifies the environment and adds the relevant description before it like "**Chapter** 3", "**Table** 2", "**Figure** 5a", depending on the environment. This saves time and mistakes.

2.3.3 chemformula

`chemformula` is useful for chemical formulas (surprise, surprise). In LᴬTᴇX to do subscripts and superscripts, you often have to access `math mode`, which then formats the element letters wrong, as shown here with $H_2O_{(l)}$ (`$H_{2}O_{(l)}$`). However, we can use format it much nicer with $H_2O_{(l)}$ (`\ch{H2O_{(l)}}`). This becomes especially useful when it comes to more complicated chemical equations like Cr_2O_3 (`\ch{Cr2O3}`) and $C_6H_{12}O_6$ (`\ch{C6H12O6}`). There are more features I believe that are in the documentation.

2.3.4 siunitx

Often, you want to use units, either stating something like:

- Wm^{-2} (`\si{\W\m^{-2}}`)
- $20\,°C$ (`\qty{20}{\celsius}`)
- $500\,kHz$ to $1300\,kHz$ (`\qtyrange{500}{1300}{\kHz}`)

Note how it automatically gives a space between the number and the unit, which is the correct way to do it. You also have complicated numbers that would be a pain to write out in `math mode` such as 1.1×10^5 (`\num{1.1d5}`). You can also define your own units so that you can write them out quickly like "mbar", which is not in the package but useful to have on hand.

There are so many options with `siunitx` that I think it is worth looking at the documentation page to get inspired!

2.3.5   spreadtab

`spreadtab` turns tables in LaTeX into Excel spreadsheet like tables in capability. I used this during my thesis to have a word count table for each chapter to best calculate the overall word count of the document without requiring an external file. The table was based on my word count table.

2.3.6   listings

I used `listings` for inputting code into the document, and you can do syntax-based formatting too. Examples of this can be found in this post.

```matlab
function [y, y_total] = customFunc(x)
% This is an example MATLAB function.

y = x.^2;
y_total = sum(y);
fprintf("The total is %.2f\n", y_total);
```

2.4   Misc.

2.4.1   pifont

`pifont` allows you to have more symbols in your work. I found the following useful ✗ (`\xmark{}`) ✓ (`\cmark{}`). These are user made commands which make use of `\ding{51}` which is part of `pifont`. See documentation especially "Table 2" which has the list of symbols available via ding.

2.4.2   dirtytalk

Useful for when you want "nice" (`\say{nice}`) quotations rather than "worse" (`"worse"`) quotations. More of a personal preference. Note that it does not work in figure captions.

2.4.3   todonotes

`todonotes` is the nicest workaround for adding Word-like comments into your thesis (`\todo[color=yellow]{example}`) for a supervisor to read he may not be   example

Overleaf savvy. An example is given above but read the documentation on the more variety of ways you can add them in.

2.4.4 lipsum

lipsum is great for making templates or filler text by generating random Latin text.

2.4.5 Quality of Life Packages

Some of the packages added are for quality of life when using Overleaf/LATEX. A great example is silence which can turn off the "underfill" warnings which pop up. The warning about pages containing just floats (e.g. just figures and no text) has been turned off too.

Appendices

A
Numbers & Letters

Contents

A.1 These are some Numbers and Letters

A word looks like this, and

$$1 + 1 = 2$$

[1].

B

Word and Page Count

B.1 Word count

```
File: Oxford_Thesis.tex
Encoding: ascii
Words in text: 1665
Words in headers: 67
Words outside text (captions, etc.): 75
Number of headers: 33
Number of floats/tables/figures: 4
Number of math inlines: 4
Number of math displayed: 1
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  0+4+0 (1/0/0/0) _top_
  0+1+0 (1/0/0/0) Chapter: Introduction
  25+2+0 (1/0/0/0) Section: A section
  30+2+11 (1/1/0/0) Subsection: A subsection
  0+2+0 (1/0/0/0) Section: Why \LaTeX{}?
```

```
155+7+0 (1/0/0/0) Subsection: Working At A Chapter At A Time

76+3+0 (1/0/0/0) Subsection: Citations \& Reference Managers

89+2+0 (1/0/0/0) Subsection: Packages \& Commands

125+6+0 (1/0/0/0) Chapter: How To Use This Template \& \LaTeX

88+1+0 (1/0/0/0) Section: Drafting

0+1+0 (1/0/0/0) Section: Figures

89+1+23 (1/1/0/0) Subsection: subcaption

255+1+33 (1/1/0/0) Subsection: inkscapelatex

42+2+8 (1/1/0/0) Subsection: pdflscape \& makecell

11+1+0 (1/0/0/0) Subsection: pgf-pie

100+1+0 (1/0/0/0) Subsection: placeins

0+1+0 (1/0/0/0) Section: Text

48+1+0 (1/0/0/0) Subsection: Acronyms

67+1+0 (1/0/3/0) Subsection: cleverref

77+1+0 (1/0/1/0) Subsection: chemformula

109+1+0 (1/0/0/0) Subsection: siunitx

51+1+0 (1/0/0/0) Subsection: spreadtab

36+1+0 (1/0/0/0) Subsection: listings

0+1+0 (1/0/0/0) Section: Misc.

49+1+0 (1/0/0/0) Subsection: pifont

25+1+0 (1/0/0/0) Subsection: dirtytalk

51+1+0 (1/0/0/0) Subsection: todonotes

14+1+0 (1/0/0/0) Subsection: lipsum

47+4+0 (1/0/0/0) Subsection: Quality of Life Packages

0+2+0 (1/0/0/0) Chapter: Numbers \& Letters

6+6+0 (1/0/0/1) Section: These are some Numbers and Letters

0+4+0 (1/0/0/0) Chapter: Word and Page Count

0+2+0 (1/0/0/0) Section: Word count


(errors:6)
```

Overall

	Words	Date	% of 100	% of thesis
Introduction	41	XX/YY	41	77.4
Conclusions	6	XX/YY	6	11.3
Appendix	6	XX/YY	6	11.3
Sum	53		53	

B.2   Page count

Total page count: 18.

References

[1]   Alfred North Whitehead and Bertrand Russell. *Principia mathematica to* 56.*
      Vol. 2. Cambridge University Press, 1927.