Accessible PhD Thesis Template

Your Name

14th November 2026

Abstract

This document provides an accessible LaTeX template for PhD theses in mathematics. It is designed to help students meet Lancaster University's requirements on digital accessibility while keeping the writing process simple and close to standard LaTeX practice.

The template uses the new PDF tagging system of LaTeX (available in recent TeX Live distributions), which enables the production of tagged PDF/UA–2 documents with improved support for screen readers. Common elements of a thesis such as headings, lists, figures, tables, mathematical expressions, theorems and code examples are preconfigured to work as accessibly as current tools allow.

The aim of this template is not to make students experts in accessibility, but to ensure that an accessible thesis can be produced without additional effort. Students can simply use the provided environments in the usual way, following the brief guidance included in Chapter 1. Where full accessibility is not yet possible (for example, with complex tables or highlighted program code), practical alternatives are given. In particular, practical advice is given on how to use this template and how to deal with typical issues that may arise during compilation.

This template may be used as a starting point for writing a PhD thesis or as a reference for including accessible mathematical content in other documents.

Updated: Mai 2026 (minor adjustments)
Created: February 2026

Acknowledgements

I thank ...

Contents

Abstract i

Acknowledgements iii

1 Instructions for accessibility 1

1.1 What is the purpose of this document? . . . 1

1.2 A note on using the template . . . 1

1.3 What is accessibility? . . . 2

1.4 Why does the default LaTeX-to-PDF create inaccessible PDFs? . . . 2

1.5 Advice how to create (almost) accessible pdfs with LaTeX . . . 3

1.5.1 Cross-references . . . 4

1.5.2 Tips for accessible mathematics . . . 4

1.5.3 Bibliography and citations . . . 5

1.5.4 Common pitfalls to avoid . . . 5

1.6 Practical workflow for accessible compilation . . . 6

1.7 Examples of some code . . . 8

1.7.1 Equations . . . 8

1.7.2 Definitions, Theorems and more . . . 9

1.7.3 Enumerates and itemize . . . 10

1.7.4 Figures 10
1.7.5 Tables 11
1.7.6 Program code 11
1.8 Create accessible documents beyond pdf 13

2 Introduction **15**

List of Figures

1.7.1 Illustration of $A \cup B$. 11

1.7.2 Illustration of $A \cup B$, $A \cap B$, $A \setminus B$, A^c. 12

List of Tables

1.7.1 An example of an accessible table 11

This page intentionally left blank

Chapter 1

Instructions for accessibility

1.1 What is the purpose of this document?

According to the MARP (Manual of Academic Regulations and Procedures for Postgraduate Research), the formatting of PhD thesis must follow Lancaster University's guidance on accessible documents and resources (see Appendix 2, Point 8). This document is a template you can use for your PhD thesis to fulfil this requirement, and also it gives you some useful advice and information.

Further this document is created using the relative new document tagging system. It is created so that it runs under pdflatex and under LuaLaTeX, and also under MikteX and TeXLive. From an accessibility point of view, LuaLaTeX together with a recent TeX Live installation is the recommended way.

However, it is important that you are using the latest version of LaTeX

This file was created on 30.5.2025 and updated on 22.2.2026 by Dirk Zeindler. If you have any comments, suggestions or corrections then please contact the disability officer.

1.2 A note on using the template

This template is designed so that you do not need to change much about your usual way of writing LaTeX documents. Most of the accessibility features (such as tagging,

MathML, and structure) are handled automatically in the background. If you use the environments provided in this template in the standard way, you will already produce a document that is much more accessible than a normal LaTeX PDF. Only a few small rules need to be followed, which are described in this chapter.

1.3 What is accessibility?

Document accessibility refers to how easily people, including those with disabilities, can perceive, understand, navigate, and interact with its content. Making a document accessible ensures that everyone, including those with visual, auditory, cognitive, or motor impairments, can access and use its information. This is often important for legal purposes. For example, under the Equality Act of 2010, higher education providers have a duty to make reasonable adjustments that enable disabled students to access the full student learning experience.

1.4 Why does the default LaTeX-to-PDF create inaccessible PDFs?

LaTeX was created in 1984 as a typesetting language for mathematical notation and for producing professional-looking documents suitable for printing. Requirements have changed since then. In particular, there is now also a need to produce accessible PDFs, which are readable by a screen reader. The three main problems with the conversion from LaTeX to PDF are that

- that equations cannot be read by screen readers,
- it is not possible to add alternative text to images and
- PDFs are not tagged, i.e. they do not contain tags [1].

[1] Tags are embedded within the pdf and are invisible to the reader. Screen readers uses them to identify headings, paragraphs, lists, footnotes, images and tables. Also, tags are used to store additional information such as alternative text.

Therefore, from August 2020, PDFs generated from LaTeX using the default conversion to PDF should not be considered accessible. The LaTeX community is aware of this problem and is working hard to resolve these. There has been very good process in the recent year, see for instance the LaTeX Tagging Project. In particular, this document is using the new pdf tagging system, which produces pdfs that are very close to fulfill the accessibility standards.

1.5 Advice how to create (almost) accessible pdfs with LaTeX

You can find on the site Accessibility checklist for LaTeX documents some information of Lancaster University how to create accessible documents. You will see that most of the advice is either already implemented in this template or something you do anyway. Here some additional advice.

- If you would like to emphasis some text than you should use `\emph{}` instead of `\textit{}`.
- If you need to insert hyperlinks, the full web address can often be very long. In this situation, you can use the command `\href{url}{linktext}` rather than `\url`. You should avoid long raw URLs in text, and do not use 'click here' as the link text. Use a descriptive text instead, such as 'More information can be found on ASK', to convey clear and accurate information about the destination.
- For tables, it's important to use a header row, and if needed, a column header for structure and navigation. For tables that span across pages, make sure to repeat the header row at the top of each page.
- People with visual impairments often cannot perceive images or diagrams clearly, which makes it difficult for them to understand their intended meaning. Therefore, it is important to provide each image or diagram with a short descriptive text that can be read aloud by a screen reader, ensuring that the visual content remains accessible. This description is called *alternative text* (alt text). The tagging system provides a simple way to add alternative text directly to `\includegraphics`:

```
\begin{figure}
 \centering
 \includegraphics[alt={A placeholder image.}]{example-image}
 \caption{An example figure}
\end{figure}
```

For guidance on how to write effective alternative text, useful tips can be found on the Lancaster University page on adding alternative text. You can find some examples in the Section 1.7.4.

1.5.1 Cross-references

Cross-references are an essential part of an accessible document, as they help screen readers and other assistive technologies navigate the content. You should always use the commands `\label` and `\ref` (or `\eqref` for equations) to refer to numbered objects such as figures, tables, equations or theorems. For example:

```
As shown in Theorem~\ref{thm:main}, ...
See Figure~\ref{fig:example}.
From Table~\ref{tab:data} we observe that ...
```

Never type reference numbers manually.. This is error-prone, time-consuming, and breaks navigation for screen-reader users. Using `\label` and `\ref` keeps all references up to date automatically and makes the document easier to navigate in both print and digital formats.

1.5.2 Tips for accessible mathematics

The new LaTeX tagging system automatically generates semantic MathML for mathematical expressions, which greatly improves accessibility in the long term. To ensure correct results, consider the following guidelines:

- Use standard LaTeX math commands and environments, such as `equation`, `align`, or `\[...\]`.

- Avoid manually inserting line breaks inside equations; instead, use `align` or `split`.
- Write inline mathematics sparingly and keep it short.
- Do not include long pieces of text inside math mode; screen readers cannot parse such content correctly.
- Avoid inserting mathematical expressions as images. They cannot be read by assistive technologies.

At present, most screen readers cannot yet read MathML embedded in PDF files, but this is expected to improve as PDF/UA–2 support evolves.

1.5.3 Bibliography and citations

The bibliography is automatically integrated into the document's tag structure. To ensure accessibility:

- Use either `bibtex` or `biblatex`. Both approaches generate structured bibliography entries.
- Do not type references manually; this produces untagged text.
- If you include URLs or DOIs, use `\href` with descriptive link text rather than showing raw URLs.

A typical workflow is to maintain a `.bib` file and run `bibtex` or `biber` to generate the bibliography. This ensures consistent formatting and correct tagging.

1.5.4 Common pitfalls to avoid

When creating an accessible document, the following common mistakes should be avoided:

- Do not use `\textit` to emphasise text; use `\emph` instead, as it conveys semantic emphasis.

- Do not use colour alone to convey meaning. Always provide a textual description as well.
- Do not insert multiple spaces, `\quad`, or manual positioning to adjust layout. Use proper LaTeX environments instead.
- Avoid decorative images or diagrams unless you mark them as `artifact` so that screen readers skip them.
- Avoid complex TikZ overlays or highly custom drawing code, as these are not yet fully supported by the tagging system.
- Do not manually format bibliographic entries. Always use a bibliography tool such as `bibtex` or `biblatex`.
- Avoid packages that are not compatible with the tagging system, such as `listings`, `minted` and some extended verbatim packages.

1.6 Practical workflow for accessible compilation

The following recommendations address common issues that arise when working with the accessible template:

1. Enabling the command `\DocumentMetadata` significantly reduces the compilation speed of a document. For larger documents, compilation may take up to several minutes.

   It is therefore recommended to disable `\DocumentMetadata` during development (for example by commenting it out) in order to speed up compilation and simplify debugging. For the final version of the document, `\DocumentMetadata` should be re-enabled to produce an accessible PDF.

   It is advisable to enable `\DocumentMetadata` occasionally during development to ensure that no compatibility or compilation issues arise.

2. The compilation with `\DocumentMetadata` is not yet fully as robust as for a standard LaTeX document. In particular, compilation may occasionally fail even

though there is no error in the source code. Furthermore, a failed compilation can corrupt the generated auxiliary files (e.g. `.aux`, `.toc`, `.out`, `.log`) and lead to additional, seemingly unrelated error messages.

If unexpected or unusual errors occur, it is therefore recommended to delete all auxiliary files and recompile the document from scratch. In particular, `\DocumentMetadata` generates an auxiliary HTML file. This is not a usable HTML version of the document and should also be removed during cleanup.

3. When switching between development mode and the accessible version (with `\DocumentMetadata` enabled), a clean compilation is recommended, i.e. to delete all auxiliary files. This avoids inconsistencies and compilation errors.

4. Not all LaTeX packages are compatible with the PDF tagging system. If unexpected errors occur when `\DocumentMetadata` is enabled, it is advisable to temporarily disable recently added packages in order to identify potential incompatibilities.

   As a first indication, external resources (including AI tools) may help to assess whether a particular package is compatible with `\DocumentMetadata`.

   This template already includes a set of commonly used packages that are known to work reliably with the tagging system. In many cases, no additional packages are needed, and it is advisable to only introduce new packages when necessary.

5. The spacing in this template is slightly more structured than in a standard LaTeX document. This improves readability and leads to a more compact and visually consistent layout.

   A disadvantage is that less vertical space is available for figures and tables. In particular, it may happen that LaTeX is not able to place a figure or table. In such cases, you may receive a message such as *Float(s) lost!*, and the figure or table may disappear from the output. In some situations, compilation may fail completely and no PDF is generated. This often occurs when using restrictive placement options such as `[h]` or `[h!]`, in particular inside structured environments like theorem environments.

   To avoid this issue, you can replace the floating environment with a non-floating construction. A standard floating figure is defined as:

```
\begin{figure}
  \centering
  \includegraphics[alt={A placeholder image.}]{example-image}
  \caption{An example figure}
  \label{fig:example_figure}
\end{figure}
```

This can be replaced by the following non-floating construction:

```
\begin{center}
  \includegraphics[alt={A placeholder image.}]{example-image}
  \captionof{figure}{An example figure}
  \label{fig:example_figure}
\end{center}
```

This ensures that the figure appears exactly at the desired location and avoids issues with LaTeX's float placement mechanism.

The same approach can be used for tables by using `\captionof{table}`.

1.7 Examples of some code

In this section, we give some examples to illustrate how things look and how they can be implemented.

1.7.1 Equations

We have

$$B(M,N) = \sum_{m,n} \alpha_m \beta_n \exp\left(2\pi i \frac{\overline{m}}{n}\right) \tag{1.7.1}$$

1.7.2 Definitions, Theorems and more

The theorem environments in this document are implemented using the `tcolorbox` package. The setup in the preamble is somewhat involved, but this has already been done for you. If you are happy with the appearance of the theorem boxes, you do not need to change anything — simply use the environments in the usual way, e.g.

`\begin{theorem}...\end{theorem}`.

A classical solution without frames is also included in the preamble, but commented out.

Definition 1.7.1

A set Ω is *countable* if one can assign to each $\omega \in \Omega$ a natural number $n \in \mathbb{N}$ so that each $n \in \mathbb{N}$ correspond to at most one $\omega \in \Omega$. If this is not possible, we call the set *uncountable*.

Example 1.7.2

The sets $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$ are countable. Furthermore any subset of $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$ are countable. However, not each set is countable. For instance, $\mathbb{R}$ and $\mathbb{C}$ are not countable.

Lemma 1.7.3

Consider an arbitrage free finite market model, realised on the probability space $(\Omega, \mathbb{P})$, and let Φ be an $\mathcal{F}_T$ measurable random variable, which is not replicable. Then there exist *risk neutral measures* $\mathbb{P}_*$ and $\mathbb{P}'_*$ such that

$$\mathbb{E}_*[\Phi] \neq \mathbb{E}'_*[\Phi]. \tag{1.7.2}$$

Theorem 1.7.4 (First fundamental theorem of asset pricing, e.g. [1, Thm 4.2.6])

A finite market model is arbitrage free if and only if there exists at least one *risk neutral probability measure* $\mathbb{P}_*$.

Corollary 1.7.5

If a finite market model is arbitrage free but not complete then there exists infinitely many *risk neutral probability measure* $\mathbb{P}_*$.

1.7.3 Enumerates and itemize

An example of a numbered list:

1. The first case.
   (a) A subcase.
   (b) Another subcase.
2. The second case.
3. The third case.

An example of a unnumbered list

- The first case.
  – A subcase.
  – Another subcase.
- The second case.
- The third case.

1.7.4 Figures

Figure 1.7.1 contains an example of a Figure with alternative text.

Figure 1.7.2 is an example of a figure with subfigures and alternative text. In particular, you can refer to the entire Figure 1.7.2 or to an individual subfigure such as Figure 1.7.2a.

Subfigures can be very useful, but they may further complicate the placement of figures and may interfere with the tagging when `\DocumentMetadata` is enabled. Therefore, they should be used with care.

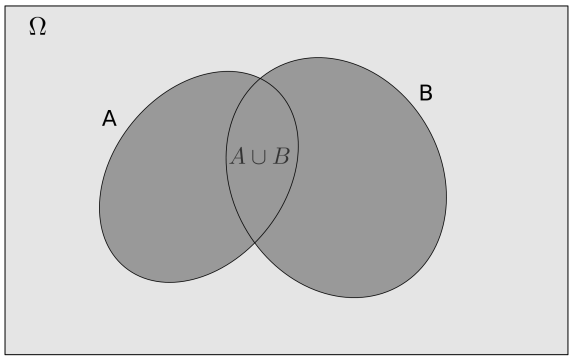


Figure 1.7.1: Illustration of $A \cup B$.

1.7.5 Tables

The new tagging system also supports accessible tables. In order for assistive technologies to interpret a table correctly, it is important that the structure is clear and that header cells are identified. This template enables table header rows globally through the option `table/header-rows=1` inside the `\DocumentMetadata` declaration.

A simple example is shown below:

Item	**Value A**	**Value B**
Apples	10	20
Bananas	15	25
Cherries	12	22

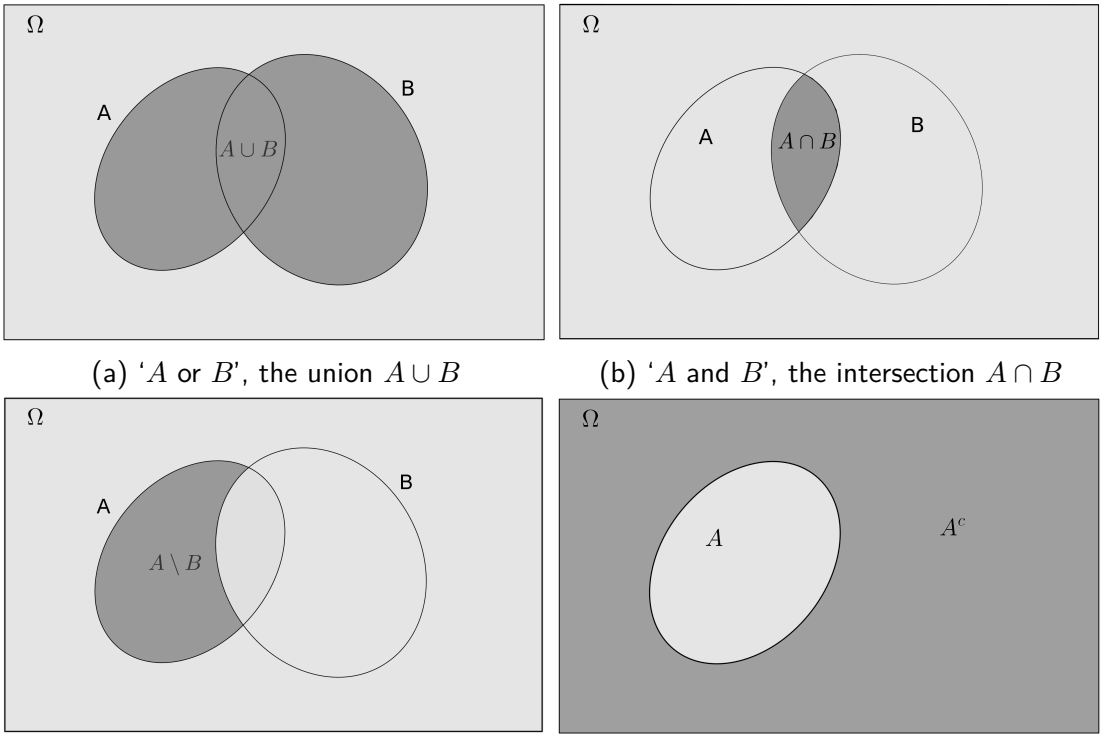
Table 1.7.1: An example of an accessible table

You can refer to this table using, for example, Table 1.7.1. Ensure that the first row contains the header cells for assistive technologies. If a table spans multiple pages, consider using the `longtable` environment, which is partially supported.

1.7.6 Program code

Including program code in a thesis can be done in various ways, but the options are limited when producing a tagged PDF/UA–2 document.

The classical `listings` package is widely used, but it is currently not compatible with the new PDF tagging system and typically results in compilation errors. Likewise, the



(a) 'A or B', the union $A \cup B$

(b) 'A and B', the intersection $A \cap B$

(c) 'A but not B', the set difference $A \setminus B$

(d) 'A does not occur', the complement A^c

Figure 1.7.2: Illustration of $A \cup B$, $A \cap B$, $A \setminus B$, A^c.

popular `minted` package produces visually attractive, colour-highlighted code, but it breaks the tagging structure when tagging is enabled, even though the document may still compile successfully.

At present, there is unfortunately no reliable method to create a code environment that is fully compatible with the tagging system and at the same time provides a caption, numbering, and cross-referencing. Properly tagged verbatim content is still being developed within the Tagged PDF project.

For this reason, this template provides a simple code environment based on `tcolorbox` and the standard `verbatim` environment. This approach is stable, preserves line breaks, and works reliably with tagged PDF output. For example:

```
for i in range(10):
print(i)
```

If you prefer not to use a frame around the code, you may simply use the `verbatim` environment on its own.

1.8 Create accessible documents beyond pdf

You can ignore this section when creating your thesis. However, for the sake of completeness, it's important to note that the difficulty of creating accessible documents isn't limited to LaTeX documents. This issue also applies to Word documents, scanned PDFs, and other documents.

You can find some information on the site Creating accessible PDFs how to create accessible pdf's with Acrobat Pro. Lancaster University has also created a Moodle course that covers Creating Accessible Resources with Word, as well as how to make scanned PDFs accessible.

This page intentionally left blank

Chapter 2

Introduction

Some text.

This page intentionally left blank

Bibliography

[1] Amanda Turner and Dirk Zeindler. *Stochastic Finance: An Introduction with Examples*. Cambridge University Press, 2023. 9