THE UNIVERSITY *of* EDINBURGH

MEng Structural Engineering with Architecture

Thesis Title

by

Author Name

2024

Declaration

This thesis is submitted in partial fulfilment of the requirements for the degree of MEng Structural Engineering with Architecture. I declare that this thesis conforms to the University's Academic Misconduct policies. It was composed by myself, the work contained therein is my own, except where explicitly stated otherwise in the text, and it has not been submitted, in whole or in part, for any other degree or professional qualification.

Author Name

Word Count: 4576 words

This thesis was conducted under the supervision of [Prof/Dr] Supervisor Name.

Abstract

The abstract of the thesis (about 300 words long).

Dedication

Your dedication. Try to keep it within one page.

Contents

List of Figures

List of Tables

Abbreviations

CIBSE Chartered Institution of Building Services Engineers.

CO$_2$ Carbon dioxide.

CO$_2$e Equivalent Carbon dioxide.

CV(RMSE) Coefficient of Variation of the Root Mean Squared Error.

E+ EnergyPlus.

EPW EnergyPlus Weather file.

Note: Author abbreviations are shown in their corresponding reference entry.

Nomenclature

Term	Description	Units
C_D	Discharge coefficient	—
μ	Mean	*as indicated*
σ	Standard deviation	*as indicated*

Chapter 1

Introduction

This LaTeX template and the instructions it contains will enable you to prepare your manuscript in an electronic format (pdf). Authors are responsible for ensuring the accuracy of all information contained in their manuscripts (e.g. proper names of organisations, data and findings, references, etc.).

LaTeX is a highly advanced type-setting system with a large number of various packages that can be loaded on demand to access certain type-setting functionalities. Overleaf is a web-service that enables you to write and type-set LaTeX documents even more easily, as (1) it does not require you to install anything on your computer, (2) it saves your document securely online; and (3) it even implements version control so that you could return to some prior version of your document if really necessary.

It is not the intention of this template to list everything possible with LaTeX. However, it will guide you for the most important type-setting issues you will definitely and/or most likely deal with. You will see that, for many of them, we in fact direct you to online resources. This is because these are already of excellent quality and are continuously maintained. It also shows that, when it comes to LaTeX, if you have a question, your best option to find the answer is to search online.

1.1 Resources

We provide some explanations in the following sections on how to create and populate a LaTeX document, but bear in mind that the following resources will soon come in handy.

LearnXinYminutes / LaTeX This gives a quick tour of LaTeX (5 to 10 minutes). It is not meant to be a comprehensive, authoritative resource, but rather an overview of how it works and how to think about it. At the end of the tour it lists very useful resources.

Overleaf's documentation This template is hosted in Overleaf, an incredibly useful service for LaTeX. The documentation is quite useful, and includes a 30-minute overview of LaTeX as well as general introduction to key ideas.

WikiBook / LaTeX An amazing online guide to LaTeX. It is well structured, comprehensive, and is easy to navigate.

Overleaf's templates It is very common to have a look around and find other people's documents doing things that you might want to replicate in your document in terms of LaTeX is used. For example, Overleaf includes a gallery that showcases things that might be helpful.

StackExchange / Tex Questions-Answers website for all things LaTeX. Very often you can just literally write what you are trying to do and it will retrieve useful answers. In fact search engines often return StackExchange pages.

CTAN CTAN is the Comprehensive TeX Archive Network. This is the main repository of all the packages that are available in LaTeX. Packages tend to have great documentation. For example:

> **hyperref** This is the package we use for cross-referencing and hyperlinks in this template.
>
> **siunitx** This package supports typesetting values with units.

1.2 File Structure

Documents prepared and compiled with LaTeX contain in fact a number of files. The general principle is to dissociate content from formatting (aka typesetting) information — following a similar idea to web technology where HTML files normally contain content information while CSS files contain formatting information. The main file types are:

- `.tex` files: these files contain the content of your thesis. Not all the content has to be in one single file. It is totally fine to split it in multiple `.tex` files, for example one per chapter. In this case, one `.tex` file acts as the main document and the other ones are called within it, using the function `\include{...}`. In this template, the main document is the file `thesis.tex` and, while most of the content is within it, the front matter content is in the `frontmatter.tex` file which is included in `thesis.tex`. Similarly, the acronyms and nomenclature entries are contained in the `definitions.tex` file, and that file is called inside `frontmatter.tex` (albeit with a different mechanism). More information about the front matter is found in Section 1.3.1. Note that some typesetting information may be provided to `.tex` files, but this should be for adhoc typesetting needs not covered in the other files described below.

- `.sty` files: these files define what are called *packages* that define typesetting. First, as you can see at the top of the `thesis.tex` file, this thesis template uses the 'report' `documentclass`. That document class already comes with a number of pre-defined typesetting definitions. but, these can be altered or augmented with additional typesetting definitions by loading *packages*. Packages come in two forms; they are either:

  - Ad-hoc `.sty` files that are added to your latex project. For example, this thesis template includes the `UEDIN_SoE.ply` package; or
  - Existing packages that can simply be loaded in `.tex` or `.ply`. Existing packages are loaded using the `\usepackage{...}` function and you can see at the beginning of the `thesis.tex` file that this thesis template already loads a dozen packages. While we think these packages should already enable you to do most things you may wish to do in a thesis (e.g. add URLs or having figures with sub-figures), you may have additional needs. For this, simply search the resources listed in Section 1.1.

- `.bib` files: these are what are called Bibtex files. Bibtext is a open standard file format which is used to describe lists of references. This template document contains the `thesisReferences.bib` file that contains the references that you wish to cite in the thesis. The file is called at the end of the `thesis.tex` file (because the list of references goes at the end of the thesis), with the command `\bibliography{thesisReferences}`.

More information about how to add content to `thesisReferences.bib` and cite those references in your thesis is provided in Section 1.9.

- Graphics/Image files: these are the graphics and images you wish to add to your thesis as figures. LATEX supports many grapchics and image file formats, including: `.png`, `.jpeg` (`.jpg`), `.pdf` and `.eps`. You are most certainly familiar with the first three, but likely not the last one. In fact, one should distinguish them in two groups:

    - *Raster formats* (`.png` and `.jpeg` (`.jpg`)): these file formats store the graphic information as an array of pixels. This means that their file size grows with the number of pixels. These formats are recommended for storing photos/pictures or other types of graphics when the second group of formats below cannot be used or is not available.

    - *Vector formats* (`.eps` and `.pdf`): these file formats store the graphic information in a vector format. For example, instead of storing a black line as a series of pixels, it will store it as a line object with a start and end point and a black colour. They also store text as actual text. These formats are ideal to store graphs, diagrams and similar graphic objects. Note that using these formats normally lead to smaller file sizes (the image itself AND your final thesis), and they respond much better to graphic sizing in the template (e.g. adjusting text font size in accordance with the scaling of the image in the document). Importantly, note that if you take a graphic in raster format and save it in a vector format, it will not magically change the pixels into smart graphic objects. Any content in raster format will simply be re-included as raster content in the file.

More information about how to generate vector graphic content and adding figures to your document is provided in Section 1.4.

1.3   Text Structure

The thesis is composed of each chapters that are themselves composed of sections, sub-sections, etc. To create such a heading structure, you can use the commands `\chapter{}`, `\section{}`, `\subsection{}`, `\subsubsection{}`, and `\paragraph{}`. Note that, while `\section{}` should be inside a `\chapter{}`, a `\subsection{}` inside a `\section{}`, etc.,

a `\paragraph{}` is not numbered and can thus be added inside any of those.

If you wish to cross-reference a heading elsewhere in the text, you will need to do two things. First you must give that heading a *label* using the command `\label{}`, e.g. `\label{sec:TextStructure}`. This label should be put right after the heading declaration: either on the same line or the next one. Then, you can cross-reference the heading in the text using commands like `\cref{}`. More information about cross-referencing can be found in Section 1.8.

1.3.1   Front matter

The front matter of your thesis contains the title page and a number of sections, some mandatory, others optional. In order, these are:

Title page: This is generated automatically by LaTeX. To ensure it is complete, you must just enter your thesis title in the command `\title{}` and your full name in the command `\author{}`, both located at the beginning of the file `thesis.tex`.

Declaration: This is generated automatically by LaTeX. To ensure it is complete, you must just enter your full name in the command `\author{}` (as already explained for the title page) and your supervisor's name in the command `\supervisor{}`, both located at the beginning of the file `thesis.tex`. The text content of the declaration is defined in the corresponding environment located in the file `\frontmatter.tex`. However, you should not need to alter it. Also, note that the declaration page also adds your thesis word count automatically.

Abstract: This is mandatory. Your abstract should be in the region of 300 words long, containing 6 to 8 sentences covering the following:

1. The scientific domain and the problem within the domain which is the subject matter of the thesis;
2. The research question to be answered in the thesis;
3. The means and methods (scientific tools) used to obtain the answer to the stated research question;
4. The answer to the research question; and

5. The meaning and importance of the answer and the results obtained,

The text content of the abstract must be entered in the corresponding environment located in the file `\frontmatter.tex`.

Dedication: This is optional, but commonly used by the students to thank various people for their support. The text content of the dedication must be entered in the corresponding environment located in the file `\frontmatter.tex`.

Table of Content: This is generated automatically by LATEX.

List of Figures: This is generated automatically by LATEX.

List of Tables: This is generated automatically by LATEX.

Abbreviations: This is required if you use abbreviations, in particular acronyms, in your thesis (which is very likely). All acronyms must be defined in the file `definitions.tex`, as illustrated in it.

Nomenclature: This is optional but welcome if you use many mathematical variables throughout your thesis. All terms/variables must be defined in the file `definitions.tex`, as illustrated in it.

1.3.2   Lists

1. Numbered lists should be presented using the environment `enumerate` like in this example. This environment applies the numbering and defines the format and spacing automatically.
2. Add the option `[noitemsep]` to the environment to have no additional space between numbered list items (like is done here).

- Bulleted lists should be presented using the environment `itemize` like in this example. This environment applies the numbering and defines the format and spacing automatically.

- Add the option `[noitemsep]` to the environment to have no additional space between bulleted list items (this is not done here).

Label 1 Finally, description lists can be created when a label is preferred for each entry, as opposed to a bullet of a number.

Label 2 Add the option [`noitemsep`] to the environment to have no additional space between bulleted list items (like is done here).

Note that the three lists above were generated using the environments and options described within them. More information on how to create and format lists can be found at https://www.overleaf.com/learn/latex/Lists.

1.4 Figures

1.4.1 Simple Figure

The lines below show an example of LATEX code to create an environment that displays the image `graph_a.pdf`, gives it a caption and centres everything. It also gives the figure a label so that it can be easily cross-referenced in the text (see Section 1.8. The figure produced by that code is shown as Figure 1.1.

```
\begin{figure}[!ht]
  \centering
  \includegraphics[scale=0.7]{graph_a.pdf}
  \caption{An example graph in PDF format.}
  \label{fig:SimpleFigure}
\end{figure}
```

Note that you normally have to give the full path of an image in the LATEX project tree in order for it to be retrieved and loaded successfully. To keep things tidy, we created a folder *Figures* in the thesis project file structure and added the line `\graphicspath{{Figures/}}` at the beginning of the `thesis.tex` file. This way, if you upload all your images in the folder *Figures*, then in the command `includegraphics{}` you only need to give the image filename, and not the full path to that image file. More information on how to create and format figures can be found at https://www.overleaf.com/learn/how-to/Including_images_on_Overleaf.
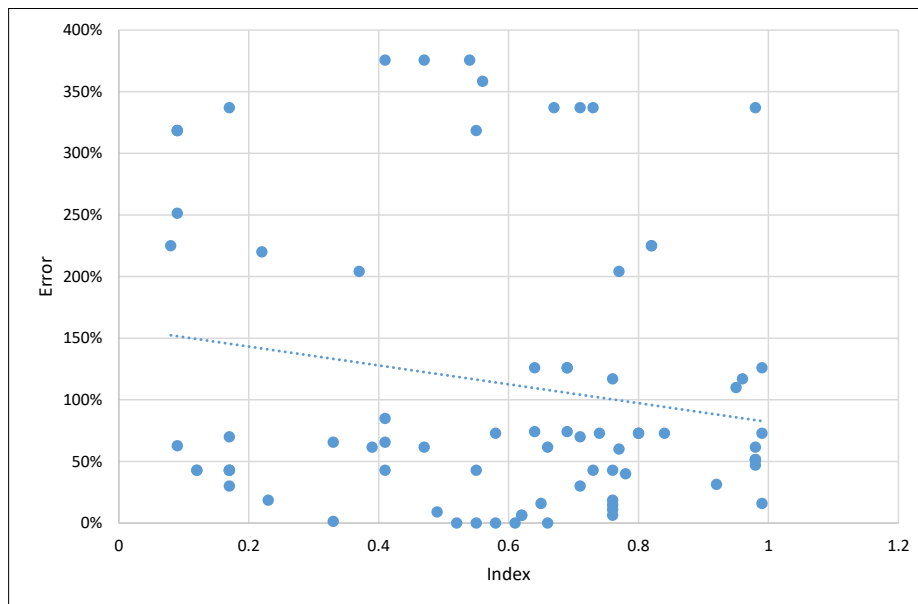
Figure 1.1: An example graph in PDF format.

1.4.2 Raster vs Vector format

Figure 1.2 shows what appears to be twice the same figure. However, the top one is rasterised (i.e. it is a grid of pixels) and is 83Kb in size, while the second is vectorised (i.e. canvas filled with shapes, including text) and is only 32Kb in size. The benefit of vectorised is not just the (generally) reduced size, but also that it does not pixelise when zooming on it.

WARNING: ".eps" and ".pdf" are common formats used in LaTeX for vectorised images. However, having this file type does **not** necessarily mean that the content is vectorised. For example, it is possible to save a ".png" picture of a graph into ".pdf". This does not magically transfer the picture into a vectorised image.

Useful software packages to create diagrams and vector images:

- Draw.io (opensource): create diagrams (save EPS, or PDF)

- MS Powerpoint: create diagrams (save PDF)

- Inkscape (opensource): to manipulate vector file (e.g. crop PDF image from MS Powerpoint to fit content).
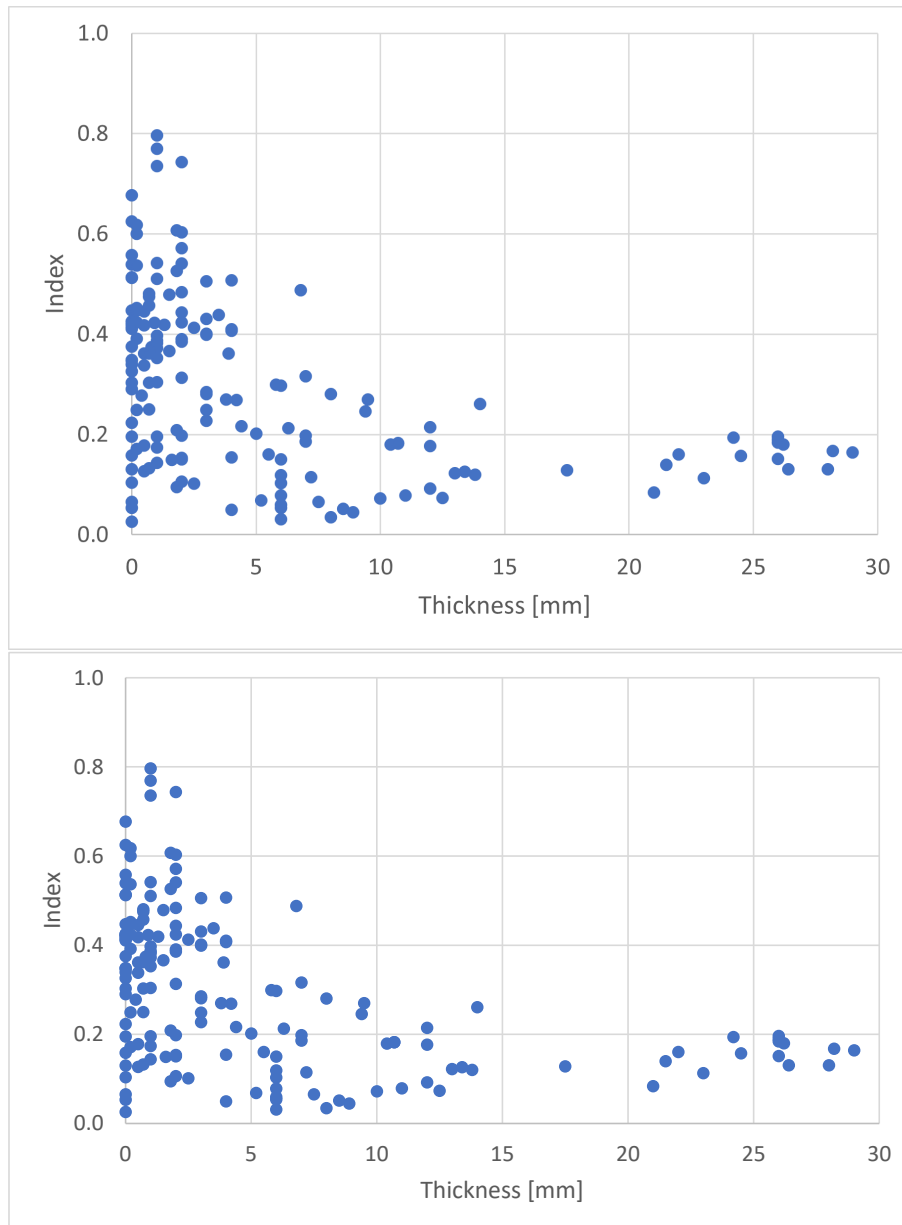
Figure 1.2: My caption text

1.4.3   LaTeX-generated figure

It is also possible to create diagrams in LaTeX using the package `tikz` (which is already loaded for you in this template). Here only the data is supplied and LaTeX code is used to create diagram/figure layout, and present the data content. The advantage of this approach is that font and font sizing is set in coherence with that of the rest of the document. However, it is quite tedious to create graphs/diagrams like that.

Figure 1.3 is an example of figure generated using the drawing package `tikz`.
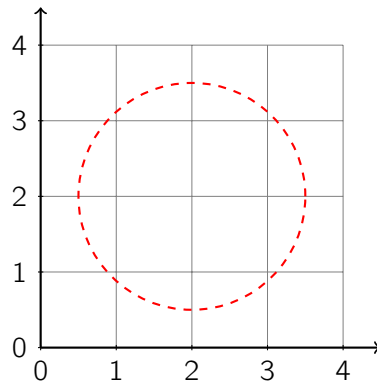
Figure 1.3: A diagram generated using the TikZ package.

1.4.4 Sub-figures

It is possible to create figures with subfigures. This requires the use of the packages `caption` and `subcaption` (which are already loaded for you in this template).

Usage is shown in Figure 1.4. You can see both subfigures have labels, which means you can cross-reference them individually in the text as Figure 1.4a and Figure 1.4b (see Section 1.8 for details on cross-referencing). More details on how to create a figure with subfigures can be found at https://www.overleaf.com/learn/latex/How_to_Write_a_Thesis_in_LaTeX_(Part_3)%3A_Figures%2C_Subfigures_and_Tables.

(a) Caption of subfigure 1

(b) Caption of subfigure 2
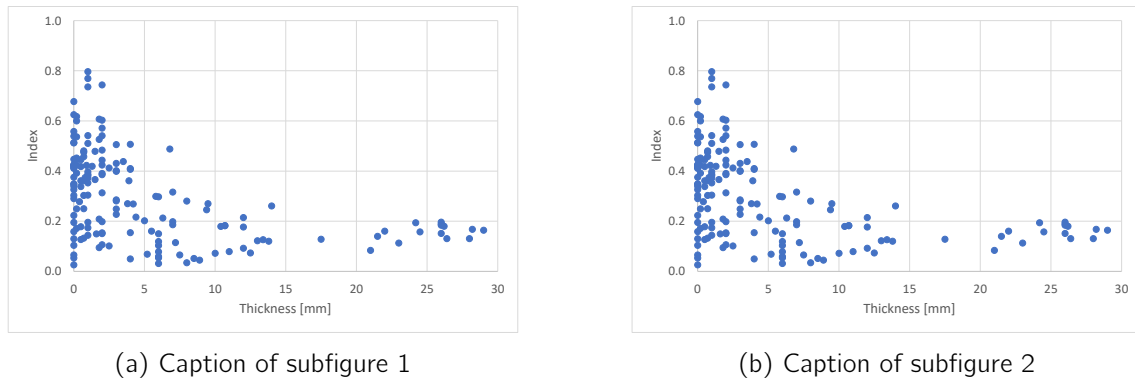
Figure 1.4: Caption of overall figure

1.4.5 And more...

Many other formatting/typesetting options exist. For example, you can put a figure in landscape mode using the environment `sidewayfigure` provided by the package `rotating`.

More information on how to create and format figures can be found at https://www.

overleaf.com/learn/latex/Inserting_Images. But, once again, if you have any specific need, we recommend you search for solutions at the locations suggested in Section 1.1 or by using your favourite search engine.

1.5 Tables

1.5.1 Simple Table

The lines below show an example of LaTeX code to create a environment that defines a table `graph_a.pdf`, gives it a caption and centres everything. It also gives the table a label so that it can be easily cross-referenced in the text. The table produced by that code is also shown as Table 1.1.

```
\begin{table}[!ht]
 \centering
 \begin{tabular}{||c c c c||}
 \hline
 Col1 & Col2 & Col2 & Col3 \\
 \hline\hline
 1 & 6 & 87837 & 787 \\
 2 & 7 & 78 & 5415 \\
 3 & 545 & 778 & 7507 \\
 4 & 545 & 18744 & 7560 \\
 5 & 88 & 788 & 6344 \\
 \hline
 \end{tabular}
 \caption{Table to test captions and labels.}
 \label{tab:MyFirstTable}
\end{table}
```

In this code, the *table* environment (i.e. `\begin{table}...\end{table}`) creates a floating environment of the type 'table' that can be given a caption and a label. `\centering` centres the table inside that table floating environment. Then, the *tabular* environment is the

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1.1: Table to test captions and labels.

environment that contains the description of the table. The tabular environment argument tells LaTeX the alignment to be used in each column and the vertical lines to insert. In the table above, ||c c c c|| defines a table with four columns in all of which the text is centred. It also instructs that double vertical lines should bound the table on the left and right. Inside the *tabular* environment, `\hline` creates a horizontal line through the table (you can use the command `\cline{...}` to make a horizontal line that only spans a subset of the columns). Finally, each row of data is entered with the cell content delimited with the character `&` and the end of the row marked with the characters `\\`.

1.5.2 Merging Rows and Columns

If you wish to merge some cells together horizontally or vertically, use the `\multicolumn{...})` or `\multirow{...}` commands, the latter requiring the `multirow` package (which already loaded for you in this template). Table 1.2 shows an example of table containing both merges columns and merged cells.

Table 1.2: Table with merged cells.

Col1	Col2	Col2	Col3
7		87837	787
5	7	78	5415
	545	778	7507
4	545	18744	7560
5	88	788	6344

If you have to insert a very long table, which takes up two or more pages in your document, use the *longtable* package. If you wish to have your table in landscape mode, consider using the *sidewaystable* or *landscape* packages.

More information on how to create and format tables (including how to colour rows/cells) can be found at https://www.overleaf.com/learn/latex/Tables and many other places

at the resources listed in Section 1.1 and elsewhere on the web.

1.5.3 Column Widths

As mentioned, in the tables above, ||c c c c|| defines a table with four columns in all of which the text is centred (it also instructs that double vertical lines should bound the table on the left and right). However, as illustrated in Table 1.3, the width of each column expands in order to accommodate the largest width of the content of any of its cells; there is no text wrapping.

To address this, you can replace the first c with p{0.2\textwidth} which forces the first column to have a width of no more than 20% of the text width, which triggers some text wrapping in this case. This approach is illustrated with the first column of Table 1.4. However, note that the content of the cells in that column is not centred anymore. To retain text centering, p{0.2\textwidth} needs to be prepended by >{\centering}, which is illustrated for the second column of Table 1.4.

Table 1.3: Table with fixed column widths.

Col1	Col2	Col2	Col3
1	6	87837	787
Best table in the world	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1.4: Table with fixed column widths.

Col1	Col2	Col2	Col3
1	6	87837	787
Best table in the world	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

1.5.4 Colouring Cells

You may wish to colour cells in a table you created. This can be done by loading the package xcolor (already loaded for you in this template), and using the cellcolor{...} command

in the cell to be coloured. This is illustrated in Table 1.5.

It is also possible to colour an entire column at once. This requires defining a new column type. Table 1.6 illustrates this with the new column type G define in the preambule of this LATEX document as:

```
\newcolumntype{G}{>{\centering\columncolor{blue!20!white}}p{0.2\textwidth}}
```

Table 1.5: Table with coloured cell.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1.6: Table with coloured column.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

1.5.5 Multipage Table

When typesetting a table, you may come across a table whose content makes it span more than page. LATEX does not handle these situations, by splitting the table automatically across two (or more) pages. Instead, you have to change your table environment to a `longtable` environment and provide the necessary information whether and how to repeat the table caption on each page, repeat the column headers, etc. How this is achieved is illustrated with Table 1.7.

Table 1.7: A sample long table.

First column	Second column	Third column
One	abcdef ghjijklmn	123.456778
	Continued on next page	

Table 1.7 – continued from previous page

First column	Second column	Third column
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778

1.6 Mathematical Expressions

1.6.1   Inline vs Display Mode

Mathematical expressions, including formulas, can be entered in two different ways: *inline* mode or *display* mode. Inline mode means that the mathematical expression will appear within the text; display mode means that it will appear in a separate line (or set of lines).

You can use two types of delimiters to typeset mathematical expressions in inline mode, `\(...\)` or `$...$`, or the *math* environment `\begin{math}...\end{math}`. Two example of inline mathematical expressions are $d = \sqrt{x^2 + y^2}$ and $M = A + B$.

You can use the delimiters `\[...\]` or the *displaymath equation* or *align* environments `\begin{equation} ...\end{equation}` to typeset mathematical expressions in display mode.

An example of mathematical expression in display mode is:

$$d = \sqrt{x^2 + y^2}$$

An example of mathematical expression in display mode is:

$$d = \sqrt{x^2 + y^2}$$

An example of mathematical expression in display mode is:

$$d = \sqrt{x^2 + y^2} \tag{1.1}$$

1.6.2   Mathematical Symbols and Greek letters

You can use an extremely broad range of Greek letters and mathematical symbols, such as:

- Binary operators: $\times \otimes \oplus \cup \cap$

- Relation operators: $< > \subset \supset \subseteq \supseteq$

- Greek letters: $\alpha\beta\gamma\rho\sigma\delta\epsilon$, and $AB\Gamma\Sigma\Delta E$

- Other: $\int \oint \sum \prod$

A first list of mathematical symbols can be found at https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols with an even more detailed list of

symbols available at https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols

1.6.3 Equation

LATEX is extremely powerful in general, but in particular when it comes to typesetting mathematical expressions, such as equations. The equation below is just one example. As you can see,

$$
\begin{aligned}
S(\omega) &= \frac{\alpha g^2}{\omega^5} e^{[-0.74\{\frac{\omega U_\omega 19.5}{g}\}^{-4}]} \\
&= \frac{\alpha g^2}{\omega^5} \exp\left[-0.74\left\{\frac{\omega U_\omega 19.5}{g}\right\}^{-4}\right]
\end{aligned}
\tag{1.2}
$$

1.6.4 And more...

Further details about typesetting mathematical expressions can be found at https://www.overleaf.com/learn/latex/Mathematical_expressions.

1.7 Properly Typesetting Units

In any scientific and engineering piece of work units play a major part. Without them our numerical information is practically useless. We need to be able to properly typeset them as in publishing they often do follow specific rules that sadly most of the time simple word processing packages ignore. Units for example are not set in italics or in random fonts that we seem to be using.

There are many LATEX packages you can use and explore to help you typeset units properly but in this template we use the `siunitx` package.

For example, `\unit{kg.m.s^{-1}}` produces $\mathrm{kg\,m\,s^{-1}}$. `\unit{\kilogram\metre\per\second}` produces the same output $\mathrm{kg\,m\,s^{-1}}$.

If you want to nicely refer to an angle you can just type `\ang{10}` to get $10°$ or `\ang{3;23;5}` for $3°23'5''$ or type `\qty{3}{mm}` for $3\,\mathrm{mm}$ or more explicitly `\qty{3}{\milli\metre}` for the same effect as $3\,\mathrm{mm}$. `\qtylist{5;10;15}{\metre}` will produce nicely $5\,\mathrm{m}$, $10\,\mathrm{m}$ and $15\,\mathrm{m}$. `\qtyproduct{5 x 10 x 15}{\metre}` produces $5\,\mathrm{m} \times 10\,\mathrm{m} \times 15\,\mathrm{m}$.

The options are very comprehensive, and no matter how complex or simple your units are

they will be typeset properly and beautifully. Detailed instructions on how to use the various commands that the `siunitx` package makes available for typesetting units properly are given in https://mirror-hk.koddos.net/CTAN/macros/latex/contrib/siunitx/siunitx.pdf

1.8   Cross-referencing

Your work will have internal references to various objects like equations, figures, chapters, sections, etc. Let's have a look at some examples.

Using the readily available command `\ref{}`, you can automatically reference such objects. For example:

- `Section~\ref{sec:Figures}` typesets 'Section 1.4'

- `Figure~\ref{fig:SimpleFigure}` typesets 'Figure 1.1'

- `Table~\ref{tab:MyFirstTable}` typesets 'Table 1.1'

- `Equation \ref{eq:sedov}` typesets 'Equation 1.2'

However, `\ref{}` requires you to actually type the type of object referenced, i.e. `Section` in `Section~\ref{sec:Figures}` in order to obtain 'Section 1.4'. A smart and easier way to cross-reference is to use the `cleveref` package (already loaded for you in this template). The `cleveref` package provides the commands `\cref{}`, `\Cref{}`, `\crefrange{}`, and many more. For example:

- `\cref{sec:Figures}` typesets 'section 1.4' and `\cref{tab:MyFirstTable}` typesets 'table 1.1'

- `\Cref{sec:Figures}` typesets 'Section 1.4'

- `\cref{sec:Figures,sec:Math}` typesets 'sections 1.4 and 1.6'

- `\crefrange{sec:Figures}{sec:Math}` typesets 'sections 1.4 to 1.6'

These commands automatically identify the type of each referenced object and ensure that those types are typeset consistently throughout the document.

Although LaTeX does not require it, it is good practice to try and name labels by giving an indication of what they represent, such as `\label{sec:...}` for labels of sections and sub-sections. This way, they can be retrieved easily and not be confused.

1.9 Bibliography and Citations

Citations of works that support your arguments and are used to reference and acknowledge the original source of information that you use, discuss or develop, is very easy.

LATEX uses a specific methodology to describe and process lists of references: Bibtex (http://www.bibtex.org) . Bibtex manifest itself in your LATEX project in the form of a file with extension .bib and requires loading the package natbib to be able to process .bib files and typeset the information they contain. In the current project, the file is called thesisReferences.bib. The file has a simple text format and essentially contains a database of entries. The most common entry is for an @article but there are many more as can be seen in the example file thesisReferences.bib. What is important is that every entry you create has its own unique label that is then used to *cite* the entry appropriately in your LATEX text and create the list of references that is created automatically and included at the end of our thesis.

Citations in the text are made using the commands \citet and \citep and possibly \citeauthor or other forms of citation commands useful for more specific cases. For example, "Zhuang et al. (2021) provided a comprehensive survey in their review paper" is obtained using the command \citet{Zhuang2021ALearning}. This command creates an entry that has the authors names as part of the sentence (here used as subjects). In contrast, "a comprehensive survey is provided in (Zhuang et al., 2021)" is obtained using the command \citep{Zhuang2021ALearning}. This commands just creates an entry that is just citation to the reference. Also note that an entry in the list of references, located at the end of the thesis.

You must let LATEX know where the bib file is that contains the details of the references that you cite in you tex file. This is done by adding the command \bibliography{file.bib} in your main .tex file. In this thesis template, \bibliography{thesisReferences.bib} is added at the end of the thesis.tex file.

LATEX, in combination with its bibliography packages, can create any style of referencing that you might need and more information can be found at https://www.overleaf.com/learn/latex/Bibliography_management_with_natbib or using different packages https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex or

alternatively https://www.overleaf.com/learn/latex/Bibliography_management_with_
biblatex. However, for this thesis template, we have already selected the AGSM style (and
suggest you keep it) that is defined in the `thesis.tex` file with the command `\bibliographystyle{agsm}`.

Note: An emerging alternative to the use of *Bibtex* (used with the `natbib` package) is
emerging. It is called *Biber* (used with the `biblatex` package). We do not explain here how
to work with Biber, but it is in fact very similar to working with Bibtex. For now, we however
recommend to stick to Bibtex.

1.10 Acronyms

There are several ways of introducing acronyms. In general it is good practice to introduce
them in full on first use. For example, `\gls{CVRMSE}` produces Coefficient of Variation of
the Root Mean Squared Error (CV(RMSE)), and it shows in full because it is the first time
it is being used. Next usage should not expand as it has already been introduced, as shown
next: CV(RMSE).

Acronyms that are very familiar in the field or self explanatory may not be expanded on
any use. For example, `\acrshort{CO2e}` produces CO_2e. Some other times, the usage is
so sporadic (e.g., used in the first chapter and used again 50 pages later) that you might
want to force showing it in full regardless of previous use: `\acrfull{CIBSE}` displays Chartered Institution of Building Services Engineers (CIBSE) (this comes in handy sometimes for
captions of figures and tables to help the reader). You can even show the definition without using the acronym at all: `\acrlong{CIBSE}` produces Chartered Institution of Building
Services Engineers.

Chapter 2

Conclusion

The conclusion of your thesis.

Note that you can also add Appendices. Appendices are introduced with the command `\appendix` (see at the end of the `thesis.tex` file) and each appendix is then entered as a chapter that can then be cross-referenced, like the example Appendix A in this thesis template.

References

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. (2021), 'A comprehensive survey on transfer learning', *Proceedings of the IEEE* **109**(1), 43–76.

Appendix A

My First Appendix

The context of Appendix A.