COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Degree in Industrial Technologies

Bachelor's or Master's final project

This is the title of your project

Author
Author's Name

Supervised by
Prof. Dr. Mc Great[a]
Prof. Dr. Mc Amazing[b]
[a] Institute of Greatness
[b] Amazing University

München 2022

Thank yous

And other important information

Abstract

Abstract content

Contents

List of Figures

List of Tables

Listings

List of Algorithms

List of Symbols

Notation	Description
test	This is a test entry for glossaries

Acronyms

Notation	Description	Page List
FEM	Finite Element Method	9, 10

Chapter 1

An overview of LaTeX

In this section, a few basic tools will be presented. In following sections more advance functionality and complex tools will be showcased. **Use this as examples and to your advantage!**

IMPORTANT: this template uses LuaLaTeX, a modern LaTeX engine. You should setup your editor to use LuaLaTeX, otherwise it will not be able to generate the document. Overleaf, for example, does not change the LaTeX engine automatically, so you have to do it yourself (it is very easy!).

IMPORTANT: read the documentation of the pacakges that you will use! Also, read general documentation about LaTeX! While the following guide below explains some of the most basic and cooler topics of LaTeX, the explanations are swallow. I do not cover details nor issues that may appear and how to fix them. Some really good resources are:

- LaTeX's Wikibook
- Overleaf's LaTeX resources
- The not so short introduction to LaTeX book
- The packages manuals!
- Any searh engine.

1.1 Basics of LaTeX

1.1.1 Text styles

The following table showcases some of the more common text styles in LaTeX.

Style	Code	Ouput
Quotes	``` ``Quotes'' ```	“Quotes”
Boldface	`\textbf{Boldface}`	**Boldface**
Italics	`\textit{Italics}`	*Italics*
Emphasis	`\emph{Emphasis}`	*Emphasis*
Underline	`\underline{Underline}`	Underline
Typewriter	`\texttt{Typewriter}`	`Typewriter`
Small caps	`\textsc{small caps}`	SMALL CAPS
Mathematical	`$Mathematical^{\pi\cdot i}$`	$Mathematical^{\pi\cdot i}$
LaTeX Comments	`% Some text`	

Table 1.1: Text styles in LaTeX.

1.1.2 Structure of a LaTeX document

For this template, which is based in the `book` class, we have the following major sections:

1. `\part{}`: Parts are fully self-contained portions of information. They leave a full blank page with only the title of the part. **This is not used and not recommended!**

2. `\chapter{}`: Your normal chapters, as you can see above. We are in the “*An overview of LaTeX.*”

3. `\section{}`: Normal sections for a chapter. We are in “*Basics of LaTeX.*”

4. `\subsection{}`: Subsections. We are in “*Structure of a LaTeX document.*”

5. `\subsubsection{}`: Subsubsections. This level tends to be quite deep and will most likely not appear in the index unless we include `\setcounter{secnumdepth}{3}` in the preamble[1].

6. `\paragraph{}`: One step deeper. By default paragraphs are not numbered.

You jus have to write what you want between the `{}` for each command, and LaTeX does the rest. It typsets the titles/sections, it adds them to the table of contents and numbers them consistently!

[1]The preamble is the part before `\begin{document}`, basically, the setup section.

1.1.3 Mathematical notation

LaTeX provides several way to include symbols and write maths. The most basic way is to include mathematical notation or symbols into the text. This is known as *inline* and can be done with `$...$`. Whatever is between the $ symbols, is typeset in mathematical notation. This is an example: $2 = \frac{4}{2}$. This is produced using `$2 = \frac{4}{2}$`.

Another method is to write mathematical formulas in *display* mode, which is separated from the text. This can be done by wrapping the text in `\[...\]`. **This is not recommended** as the next method is better. Here is an example:

$$2 = \frac{4}{2}$$

Normally, the best way is to use mathematical environments. This environments will provide more functionality and generally number the equations and allows them to be labelled. Here are a few examples:

$$2 = \frac{4}{2} \tag{1.1}$$

The equation above, eq. (1.1), is produced by writing:

```
\begin{equation} \label{eq:simpleeq}
    2 = \frac{4}{2}
\end{equation}
```

Lets showcase some more environments that help us write beautiful formulas! The `\begin{array}` environment helps us write vertically aligned formulas!

$$f(t) = \left\{ \begin{array}{lcc} A_0 + A \cdot e^{-\dfrac{t - t_0}{t_d}} & for & t \geq t_0 \\ A_0 & for & t < t_0 \end{array} \right. \tag{1.2}$$

```
\begin{equation} \label{eq:abaqus-exponential-decay}
    f(t) = \left\{
    \begin{array}{lcc}
        A_0 + A\cdot e^{-\dfrac{t - t_0}{t_d}} & for & t \geq t_0 \\
        A_0 & for & t < t_0
    \end{array}
    \right.
\end{equation}
```

The `\begin{aling}` environment may be easier to use, but it has a few quirks. Read the documentation[2] for more information.

$$\begin{aligned} a_{11} &= b_{11} & a_{12} &= b_{12} \qquad (1.3) \\ a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} \qquad (1.4) \end{aligned}$$

```
\begin{align}
    a_{11}& =b_{11}&
    a_{12}& =b_{12}\\
    a_{21}& =b_{21}&
    a_{22}& =b_{22}+c_{22}
\end{align}
```

The `\begin{subequations}` allows us to have several formulas numbered into the same reference. As shown in eq. (1.5), with the first entry being eq. (1.5a).

$$\texttt{XSYMM} \equiv U1 = UR2 = UR3 = 0 \tag{1.5a}$$

$$\texttt{ZSYMM} \equiv U3 = UR1 = UR2 = 0 \tag{1.5b}$$

```
\begin{subequations} \label{eq:symmetry-bc}
    \begin{equation} \label{eq:x-symmetry-bc}
        \text{\texttt{XSYMM}} \equiv U1 = UR2 = UR3 = 0
    \end{equation}
    \begin{equation}
        \text{\texttt{ZSYMM}} \equiv U3 = UR1 = UR2 = 0
    \end{equation}
\end{subequations}
```

1.1.4 References

One of the strongest points of LATEX is its wonderful and powerful referencing system. We can reference whatever we want by putting on a "tag" with the command `\label{xxx}`. Wherever the `\label` is, it will refer to it. You can see some examples above where we refered to a few equations by their labels, which are inside the `\begin{equation}` environment. This way, LATEX knows automatically what type of thing they are referring.

The different types of references are shown in table 1.2.

[2] http://tug.ctan.org/info/short-math-guide/short-math-guide.pdf

Package	Command	Result
LaTeX	`\ref{eq:simpleeq}`	1.1
	`\pageref{eq:simpleeq}`	3
`hyperref`	`\autoref{eq:simpleeq}`	Equation 1.1
	`\autoref{fig:textstyles}`	Table 1.1
	`\autopageref{eq:simpleeq}`	page 3
`cleveref`	`\cref{eq:simpleeq}`	eq. (1.1)
	`\Cref{eq:simpleeq}`	Equation (1.1)
	`\cpageref{eq:simpleeq}`	page 3
	`\cref{eq:simpleeq,eq:symmetry-bc}`	eqs. (1.1) and (1.5)
	`\crefrange{eq:simpleeq}` `{eq:symmetry-bc}`	eqs. (1.1) to (1.5)

Table 1.2: Different reference mechanisms. **The author recommends `cleveref`!**. It is included in this template.

1.1.5 Bibliography

Bibliography management is another strong point of LaTeX! We just need to add bibliographic entries to the bibliography database, which for this template it is the `main.bib` file. Here is what such an entry can look like:

```
@book{lovecraft2016el,
    author = {Lovecraft, H. P.},
    title = {El ćlerigo malvado y otros relatos},
    publisher = {Alianza Editorial},
    year = {2016},
    address = {Madrid},
    isbn = {9788491042105}
}
```

In order to cite the entry we just have to use `\cite{}` with the entry's identifier, like so `\cite{lovecraft2016el}` [1]. We can also have multiple cites in the same command, [1, 2] (`\cite{lovecraft2016el,norton_creep}`). It is that simple! They get automatically printed in the bibliography section.

IMPORTANT: this template uses `biblatex` as the management system, which is a powerful, flexible and modern tool. Therefore, you will need to run the `biber` command to build the bibliography after the first compilation of your document; then you will have to recompile after `biber` has run. Most editors do this by default.

You can also use third-party tools like Zotero[3] to manage your `.bib` database. Most bibliography management tools are capable of dealing with `.bib` entries!

1.1.6 Tables, images and floating environments

Probably, the part of LATEX that causes the most confusion among new users, are the so called *floating envrionments*. **Tables, images, algorithms, etc are floating envrionments**. This means that **LATEX can position them where it sees fit, not where they are written by the user.** In reality, LATEX is trying to optimise your document's layout and leave as little empty space as possible.

Sooo... How do we solve LATEX moving our floating environments? Here are a few solutions:

- We don't solve it. LATEX referencing tools allow us to easily point the reader to the table, image, etc. Therefore, it is not that problematic that the *floats* may not be where we put them!
- We can ask LATEX to try to place the image where it appears in our document. This is done with the *"here"* `[h]` placement modifier, more on placement modifiers later. **This is not a definitive solution.** This will just tell LATEX to try hard to do what we are asking. There is the `[h!]` modifier, which is even stronger.
- **A really good solution is to use** `\FloatBarrier`. It comes from the `placeins` package, included in this template. `\FloatBarrier` forces LATEX to put all floating environment that have already appeared before the position where `\FloatBarrier` appears. This is very useful to force LATEX to put all floats before another section that may not be related to the topic of those floats. Here is an example:

```latex
\section{Some topic}

\begin{figure}
    XXX
\end{figure}

\begin{table}
    XXX
\end{table}

\FloatBarrier % All previous floats will appear before this point.
```

[3] `https://www.zotero.org/`

```
\section{Some unrelated topic}
XXX
```

- We can use the placement modifier `[H]` to force the float to appear *HERE*. This is provided by the `float` package. However, **this solution is not recommended!** It can lead to some wierd and nasty document layouts!

Now, how do we actually include figures, tables, etc? They all follow the same structure, here are some examples:

Figures are declared in the `figure` environment (*SHOCK!*). You can see the image rendered in fig. 1.1.

```
\begin{figure}
    \centering % Center image horizontally
    \includegraphics[keepaspectratio, trim = 1050 12 150 30, clip, width=0.5\
    linewidth, height=0.3\textheight]{Images/monoblock-material-overview-mesh.png
    }
    \caption[Overview of \glsentryname{FEM} mesh used for the final analysis.]{
    Overview of \glsxtrshort{FEM} mesh used for the final analysis.}
    \label{fig:monoblock-overview-mesh}
\end{figure}
```

The key here is `\includegraphics`, it is what loads the graphics and allows us to set its properties. The above example is rather complex, most times you do not need these many options. Nonetheless, here is what they do:

`keepaspectratio` Keeps the size ratio of the image. Very useful if you set `height` and `width` at the same time.

`trim` It allows us to trim/cut the image. It cuts X amount of pixels from the `left, bottom, right, top`. This is useful if your image is too large and you only care about a small portion of it.

`clip` Only show the trimmed image.

`width and height` Sets the maximum size with respect to the width and height. We use `\linewidth` and `\textheight` to limit the size of the image in the page by using the page's natural lengths.

Then we have `\caption`, which is what adds the text to the image. We use `\caption[]` here, to modify the text that will appear in the "List of Figures", as I do not want my acronym `FEM` to be linked there, and therefore I use

`\glsentryname` to control that. But more about acronyms and glossaries in section 1.2 Finally, we have `\label{}` is is what allows us to give an identifier to our image so that we can reference it.

Tables are fairly easy to do once we get used to their nature. It uses the `table` floating environment with another environment that allows us to type tabulated data. A basic example is given below and shown in table 1.3.

```
\begin{table}
    \centering % To center the table
    \begin{tabular}{lcr}
      \toprule
      Heading 1 & Heading 2 & Heading 3 \\
      \midrule
      Left aligned & Center aligned & Right aligned \\
      \cmidrule{2-3} % Example of a controled rule
      Some info & Some info & Some info \\
      \bottomrule
    \end{tabular}
    \caption{Example of a table.}
    \label{tab:example-table}
\end{table}
```

IMPORTANT: the `&` symbol is used as a column separator in all alignment environments!

The column alingment options for the tabular environment can be `l, c, r, m` or `p` among others. They refer to left, center, right alignment and the `m` and `p` refer to a limited size column whose vertical alignment is either centered or natural. You could use them as `m{0.3\linewidth}` for example. If you want to aling the text horizontally with `m` or `p` you would write `>{\centering\arraybackslash}m{0.3\linewidth}`. You can change the `\centering` for a `\raggedright` for a right aligned column. But this is getting too advance! One final bit of knowledge about very long tables and dynamicly sized columns. This template includes the package `xltabular`, which includes the well-known `tabularx` environment and merges it with the `longtable` environment (for tables that can span more than one page) generating its own `xltabular` envrionment. Please red the documentation of the `tabularx`, `longtable` and `xltabular` if you need to build complex tables!

Also, **there are online tools to help you generate LATEX tables from Excel sheets**. One such example (which I am not very familiar with nor endorse) is Tableconvert.

Finally, if you have `.csv` files or similar and you want to print them in your LaTeX document, you can see section 1.5, which shows how table 1.5 was automatically generated using `pgfplotstable`.

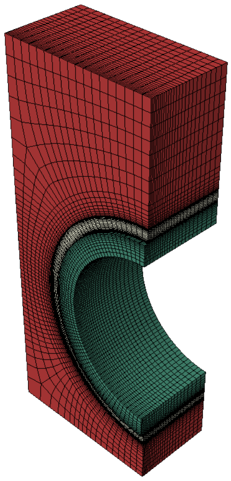

Figure 1.1: Overview of FEM mesh used for the final analysis.

Heading 1	Heading 2	Heading 3
Left aligned	Center aligned	Right aligned
Some info	Some info	Some info

Table 1.3: Example of a table.

1.2 Glossaries

Creating glossaries in LaTeX is surprisingly easy. However, it does require a bit of understanding.

For this template, the entries for the glossaries and acronyms (which are just a different type of glossary) are loaded from the `glossaries.tex` and `acronyms.tex` files. This is just to keep things organised. So, lets define some entries for our glossary and acronym list.

For glossary entries we use the `\newglossaryentry{}{...}` command. Here is an example:

```
\newglossaryentry{identifier}
{
    name={name}, % Mandatory, what gets printed
    description={description of the entry}, % Mandatory, description that appears in the glossary index
    plural={plural-name}, % Optional, in case the plural is more complex
    sort={alphanumeric entry}, % Optional, how should the entry be sorted
    symbol={\ensuremath{associated symbol}}, % Optional, prints the symbol of the entry with \glssymbol{identifier}
}
```

For acronyms, we could use the code above, but there is a simpler and more direct way of doing it with `\newabbreviation{}{}{}`. Here is how it works:

```
\newabbreviation{Identifier}{ACRONYM}{Description Of Acronym}
```

`\newabbreviation[]` supports a long set of options. For example, the `longplural={...}` option allows us to write the plural form in case it is more refined. There are many other options. The abbreviation functionality is provided by the `glossaries-extra` package.

Once your own personal entries have been created, you can use them with the following commands.

Type	Command	Result
Glossary	`\gls{test}`	test
	`\Gls{test}`	Test
	`\glspl{test}`	tests
	`\Glspl{test}`	Tests
	`\glsentryname{test}`	test
	↑ this does not produce a link	
Acronyms	`\glsxtrshort{FEM}`	FEM
	`\glsxtrshortpl{FEM}`	FEMs
	`\glsxtrlong{FEM}`	Finite Element Method
	`\glsxtrfull{FEM}`	Finite Element Method (FEM)
	`\glsentryname{FEM}`	FEM
	`\gls{FEM}`	Finite Element Method (FEM)
What?	`\gls{FEM}`	FEM

Table 1.4: Glossary and acronym types.

Wait, what happened in the second `\gls{FEM}` entry in the acronym section?

Why did it produce a different result (`\glsxtrshort{FEM}`) when compared to the first one (`\glsxtrfull{FEM}`)?

Simple: this template uses the `\setabbreviationstyle[acronym]{long-short}` style. The first time an acronym is used, it will show the full form. After that, the short form is used. All of this automatically! Isn't this magical? If you would like to show always the short form, you can delete that line from the `report.tex` or use `\setabbreviationstyle[acronym]{short-nolong}` (or any other style that you like!).

IMPORTANT: in order to show the list of glossaries and acronyms in their table of contents, you will have to run `makeglossaires`. Some editors will do that automatically for you, as they will detect you have a glossary in your document.

1.3 Automatic loading and formatting of code

1.4 Creating beautiful plots in 2D and 3D

1.5 Automatic formatting of table data

The following table, table 1.5, is formatted using the following general setup for `pgfplotstable`. The following LaTeX-`pgfplotstable` is only needed once, and it applies to "all" the automatically loaded table.

```
% Configure the general setting of pgfplotstable
\pgfplotstableset{
    every odd row/.style={
        before row={\rowcolor{gray!20}}
    },
    every head row/.style={
        before row=\toprule,
        after row=\midrule,
        % Don't print the row name or the row index!
        output empty row
    },
    every last row/.style={
        after row=\bottomrule
    },
    header=false,
    format=file,
    col sep=tab,
    search path={Data},
    font={\small}
}
```

And then the actual loading of the table. The following code setups the header (names, columns, etc) and then loads the data.

```
\begin{table}
    \newcommand{\prop}{Expansion}
    \newcommand{\propunit}{[\unit{\milli\meter\per\celsius\per\milli\meter}]}
    \centering
    \pgfplotstabletypeset[
    every head row/.append style={
        before row={
            \toprule
            \multicolumn{2}{c}{\glsentryname{Cu-OFHC}} \\
            \midrule
            \multirow{2}{\widthof{\propunit}}{\centering \prop\ \propunit} & \multirow{2}{\widthof{Temperature}}{\centering Temperature [\unit{\celsius}]} \\
            \\
        },
    },
    ]{ITER Cu You-harden for WPDIV phase II_\prop_f_T.txt}
    \caption{Automatically formatted table using \texttt{pgfplotstable}.}
    \label{tab:automatic-reading-csv}
\end{table}
```

Whats even cooler is that `pgfplotstable` uses the package `siunitx` to format the values as it is included in this template!

1.6 Some extra bits of knowledge

1.6.1 How do I prevent LATEX from splitting a word, number, etc?

LATEX will automatically break some words or numbers in order to have a nice layout of the paragraph. However, sometime we don't want that. This can be solved with `\mbox{XXX}`. This, however, may generate unexpecte behaviour! For example: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

OFHC-Cu	
Expansion [mm °C^{-1} mm^{-1}]	Temperature [°C]
$1.68 \cdot 10^{-5}$	20
$1.7 \cdot 10^{-5}$	50
$1.72 \cdot 10^{-5}$	100
$1.74 \cdot 10^{-5}$	150
$1.76 \cdot 10^{-5}$	200
$1.78 \cdot 10^{-5}$	250
$1.79 \cdot 10^{-5}$	300
$1.81 \cdot 10^{-5}$	350
$1.82 \cdot 10^{-5}$	400
$1.84 \cdot 10^{-5}$	450
$1.85 \cdot 10^{-5}$	500
$1.87 \cdot 10^{-5}$	550
$1.88 \cdot 10^{-5}$	600
$1.9 \cdot 10^{-5}$	650
$1.91 \cdot 10^{-5}$	700
$1.93 \cdot 10^{-5}$	750
$1.96 \cdot 10^{-5}$	800
$1.98 \cdot 10^{-5}$	850
$2.01 \cdot 10^{-5}$	900

Table 1.5: Automatically formatted table using `pgfplotstable`.

Bibliography

[1] H. P. Lovecraft. *El clérigo malvado y otros relatos*. Madrid: Alianza Editorial, 2016. ISBN: 9788491042105.

[2] F. H Norton. *The creep of steel at high temperatures*. McGraw-Hill, 1929. URL: `https://archive.org/details/creepofsteelathi00nort`.

Appendix A

This is an appendix