AALBORG UNIVERSITY
DENMARK

P1 Project
Mathematics-Economics

LaTeX Template

A Helping Hand For Getting Started

Authors:
Janus S. Valberg-Madsen
Janus S. Valberg-Madsen

Supervisors:
Janus S. Valberg-Madsen

October 5, 2018

AALBORG UNIVERSITY
STUDENT REPORT

Dept. of Mathematical Sciences
Skjernvej 4A
DK-9220 Aalborg Ø
http://math.aau.dk

Title:
LaTeX Template

Theme:
Project Management

Project Period:
Fall Semester 2018

Project Group:
Group XYZ

Participant(s):
Janus S. Valberg-Madsen
Janus S. Valberg-Madsen

Supervisor(s):
Janus S. Valberg-Madsen

Copies: 0

Numbered Pages: 19

Date of Completion:
October 5, 2018

Abstract:

This document serves as a LaTeX project template for first year students at Aalborg University, Dept. of Mathematics.
Examples of some common commands and environments are given to showcase their intended use.
The template can be used as-is, with students only having to write the actual contents, but interested students are encouraged to use the template as inspiration for their own customisations.

Contents

1 | First Example

This is an example chapter with content. Aside from `\chapter`, there are several levels available to use for partitioning your body text:

- `part`
- `chapter`
- `section`
- `subsection`
- `subsubsection`
- `paragraph`
- `subparagraph`

Each level is a subsection of the above level. Titles are added automatically to the table of contents. See more at `https://en.wikibooks.org/wiki/LaTeX/Document_Structure#Sectioning_commands`.

1.1 Environments

In LaTeX, you are going to be using many different kinds of *environments*. These are scopes denoted with `\begin{...}` and `\end{...}`, enclosing special content such as lists, figures, equations, etc. Table 1.1 lists some commonly used environments.

Environment	**Function**
`document`	Document contents
`table`	Floating table such as this one
`figure`	Floating figure
`equation`	Numbered equation
`align`	Aligned, multiple equations
`itemize`	Bulleted list
`enumerate`	Numbered list
`description`	Descriptive list

Table 1.1: Common LaTeX environments and their function

1.1.1 Lists

There are three essential list structures: `itemize`, `enumerate`, and `description`. The `itemize` variant produces a simple bullet list. Each item in the list are prepended by the `\item` command.

```
\begin{itemize}
  \item First item
  \item Second item
  \item Third item
\end{itemize}
```

- First item
- Second item
- Third item

The `enumerate` variant uses the same syntax for items as `itemize`, but produces a numbered list.

```
\begin{enumerate}
  \item First item
  \item Second item
  \item Third item
\end{enumerate}
```

1. First item
2. Second item
3. Third item

Finally, the `description` list in which `\item` is given an item name as an optional argument, and the contents of the line is a description of that item. This produces a list where the item names are typeset in bold followed by their descriptions as normal text.

```
\begin{description}
  \item[First item] Description of first item
  \item[Second item] Description of second item
  \item[Third item] Description of third item
\end{description}
```

First item Description of first item

Second item Description of second item

Third item Description of third item

1.1.2 Equations

One of the main reasons why people use LaTeX is the beautiful math typesetting. There are several different math environments to suit your needs, and most come in a numbered and unnumbered variants. For example, the code

```
\begin{equation}
  \label{eq:1}
  e^{i\pi} - 1 = 0
\end{equation}
```

produces the ouput

$$e^{i\pi} + 1 = 0, \tag{1.1}$$

and since it was given a label, it can be referenced with the command `\eqref{eq:1}`, which produces a clickable reference in parentheses, (1.1). If instead of `equation` you put `equation*`, the equation does not get a number. Equivalently, you can use `\[...\]`, so the code `\[ e^{i\pi} + 1 = 0 \]` produces

$$e^{i\pi} + 1 = 0.$$

If you need multiple, aligned equations, e.g. for step-by-step calculations, use the `align` environment, which aligns the contents at `&` characters. For example,

```
\begin{align*}
  (x + y)^{2} &= x^{2} + xy + yx + y^{2} \\
              &= x^{2} + y^{2} + 2xy
\end{align*}
```

produces

$$\begin{aligned}(x + y)^{2} &= x^{2} + xy + yx + y^{2}\\ &= x^{2} + y^{2} + 2xy.\end{aligned}$$

The double backslash denotes a line break. Note the asterisk; like with `equation`, `align` has both a numbered and unnumbered version. The numbered version has a seperate number for each line.

For rendering inline math, e.g. $\cos^2\theta + \sin^2\theta = 1$, use `\(...\)`. Alternatively, you can also use `$...$`, but `\(...\)` has improved spacing and error messages. See `https://en.wikibooks.org/wiki/LaTeX/Mathematics` for a good reference of symbols and commands.

1.1.3 Figures

Figures in LaTeX are input as so-called *floats* using the `figure` environment. A floating object cannot be broken over a page, so the figure will be repositioned depending on the available space on the page. The syntax is as follows:

```
\begin{figure}[placement]
  \centering
  \includegraphics[options]{path/to/image}
  \caption{The figure caption}
  \label{fig:label}
\end{figure}
```

The optional argument `placement` can be either of `h` (here), `t` (top of page), `b` (bottom of page), or `p` (put on special page with only floats). The `\centering` command is there to center the image. Among the options available for `\includegraphics`, the most important one for you will probably be `width`. To make the image take up half the page (within margins), use `width=0.5\textwidth`. All figures should have a caption, which is set with the `\caption` command, and the `\label` lets us reference it (for example, `Figure \ref{fig:me}` becomes Figure 1.1). For more info, see `https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions`.

Figure 1.1: A picture of me responding to emails from my students

Figures need not necessarily consist of images like JPG, PNG, or PDF files, but can also consist of LaTeX code. An important example of this is *TikZ*, a native vector graphics framework in LaTeX. A TikZ picture is enclosed in the environment `tikzpicture`, which can be enclosed in a `figure` environment, just like images:

```
\begin{figure}[placement]
  \centering
  \begin{tikzpicture}
    % tikz code goes here ...
  \end{tikzpicture}
  \caption{The figure caption}
  \label{fig:label}
\end{figure}
```

An example of a graph drawn with TikZ code is shown on Figure 1.2.

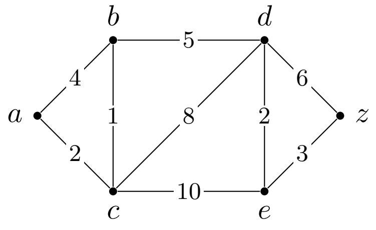

Figure 1.2: Example graph from [Rosen, 2013].

To learn TikZ syntax, see [Crémer, 2011], the tutorial sections of [Tantau, 2015], and `https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ`.

1.1.4 Tables

Like with figures, tables are also input as floats, using the `table` environment. The actual contents of a table is placed in a `tabular` environment, where columns are seperated by a `&`, and rows are seperated by a `\\`:

```
\begin{table}[placement]
  \begin{tabular}{columns}
    % table code goes here...
  \end{tabular}
  \caption{The table caption}
  \label{tab:label}
\end{table}
```

The `placement` argument is the same as for figures. The `columns` argument is a specification of the columns and consists in its most basic form of the letters `l`, `c`, and/or `r`, optionally interspersed with `|` where you want vertical lines. For example, having `\begin{tabular}{r|cc}` would tell LaTeX that the table has three columns, the first of which is **r**ight aligned, followed by a vertical line, and then two **c**enter aligned columns with no divider between them.

For a complete example of a table, see the source code for Table 1.1 (located in `fig/tab/my-table.tex`). For more information and advanced column options, see `https://en.wikibooks.org/wiki/LaTeX/Tables`.

2 | Second Example

Here is another example input file.

2.1 Custom Environments and Commands

While LaTeX provides commands for many different purposes, you will often find yourself defining your own. For this template, I have included some examples of custom environments and commands in the preamble (`premable.tex`).

Such commands can save you a lot of typing when working on a long, modular document such as a semester project. For instance, instead of typing `\mathbb{N}` every time you want the symbol for the set of natural numbers, define a shorter command, like `\N`. The syntax for defining commands is as follows:

```
\newcommand{name}[num]{definition}
```

where `name` is the command name, e.g. `\N`, `num` is the number of arguments the command takes (omit the square brackets if the command takes no arguments), and `definition` is the output of the command, e.g. `\mathbb{N}`.

2.1.1 Definitions, Theorems, Proofs

In a mathematics project, you are going to be including mathematical definitions, propositions, lemmas, theorems, etc. The `amsthm` package provides a simple way to define such environments:

```
\newtheorem{name}{Printed output}[numberby]
```

A few examples are included in the preamble. See `https://en.wikibooks.org/wiki/LaTeX/Theorems` for more information.

Definition 2.1 (Rational numbers)**.** A real number r is called *rational* if there exist integers p and q with $q \neq 0$ such that $r = p/q$ (a real number which is not rational is called *irrational*). The set of rational numbers is denoted with $\mathbb{Q}$.

Theorem 2.2 (Example of Named Theorem)**.** $\sqrt{2}$ *is irrational.*

Proof. Suppose $\sqrt{2} \in \mathbb{Q}$ and let $k = \min\{\, N \in \mathbb{N} : N\sqrt{2} \in \mathbb{N} \,\}$. But then $k(\sqrt{2} - 1) = (k\sqrt{2} - k) \in \mathbb{N}$, and $k(\sqrt{2} - 1) < k$, which is a contradiction. Therefore, $\sqrt{2} \notin \mathbb{Q}$. $\square$

2.2 Citations

When you use other people's work, you must cite them. Citations in LaTeX is usually handled with BibTeX, a procedure that identifies resources from a bibliography file you provide. Literature resources must be defined in a `.bib` file and included with the `bibliography` command. Each entry in this file must have a resource type and a key to identify it by. The syntax for citing a resource is

```
\citep[scope]{bibkey}
```

where `bibkey` is the unique identifier you give the resource in the `.bib` file, and `scope` can be used to refer to a specific chapter or range of pages.

When you cite a resource, that resource is automatically added to the literature list, and in the PDF file the citations become clickable links pointing to that list. For example, by citing [Edwards and Penney, 2014, pp. 104-110], an entry for it is added to the bibliography on the last page.

There are several different types of BibTeX entries, and each type has its own set of mandatory and optional fields. For more information, see `https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#BibTeX`.

Appendices

A | Pseudocode

To include pseudocode in your project, use the `algorithmic` environment (place inside an `algorithm` environment for figure-like floating):

```
\begin{algorithm}
  \begin{algorithmic}
    % pseudocode here...
  \end{algorithmic}
  \caption{algorithm title}
  \label{alg:label}
\end{algorithm}
```

Algorithm 1 Bubble Sort

```
procedure BUBBLESORT( $a_1, \dots, a_n$ : real numbers with $n \geq 2$ )
    for $i \coloneqq 1$ to $n - 1$ do
        for $j \coloneqq 1$ to $n - i$ do
            if $a_j > a_{j+1}$ then swap $a_j$ and $a_{j+1}$
                                        ▷ $a_1, \dots, a_n$ are now in increasing order
```

See `https://en.wikibooks.org/wiki/LaTeX/Algorithms#Typesetting_using_the_algorithmicx_package` for a quick reference of the syntax.

B | Python Source Code

If you implement algorithms in specific programming languages, you might want to include the source code in an appendix like this one. The package `listings` provides funtionality for including syntax highlighted source code for many different languages.

With the environment `lstlisting` you can type code directly, but most often you will probably need the `\lstinputlisting` command, which lets you input the code from a source file, i.e.

```
\lstinputlisting[options]{path/to/source_file}
```

Some notable options for this command are:

caption sets the listing caption

label sets the listing label

language sets the language for the syntax highlighter locally

style sets the highlighter style locally

firstline starts the listing at the specified line number

lastline ends the listing at the specified line number

In the preamble, you can also set options globally with `\lstset`. For example, if all your source code is in Python, you might put `\lstset{language=Python}`.

```python
def bubblesort(a):
    for i in range(len(a)-2):
        for j in range(len(a)-i-1):
            if a[j] > a[j+1]:
                tmp = a[j]
                a[j] = a[j+1]
                a[j+1] = tmp


ex = [54, 26, 93, 17, 77, 31, 44, 55, 20]
bubblesort(ex)
print ex
```

Listing B.1: Bubble Sort in Python

For more information, see `https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings`.

List of Figures

List of Tables

Bibliography

Crémer, J. (2011). *A very minimal introduction to TikZ*. Toulouse School of Economics. `https://cremeronline.com/LaTeX/minimaltikz.pdf`.

Edwards, C. H. and Penney, D. E. (2014). *Calculus: Early Transcendentals*. Pearson, 7th edition.

Rosen, K. H. (2013). *Discrete Mathematics and Its Applications*. McGraw-Hill, 7th edition.

Tantau, T. (2015). *The TikZ and PGF packages*. Institut für Theoretische Informatik, Universität zu Lübeck. `http://mirrors.dotsrc.org/ctan/graphics/pgf/base/doc/pgfmanual.pdf`.