Workshop LaTeX

Gijs Hendrickx

Het Gezelschap "Practische Studie"

February 19, 2020

Contents

1 Introduction

This small booklet is a starting point for using LaTeX as support to the workshop. The main focus will be on writing a report, but other document styles are highly comparable; e.g. an article.[1] The online support for LaTeX is huge and almost everything can be found on the Internet. Therefore, if you want to do something that is not included in this document – which over time you will – just use the Internet and use the keyword `latex`.

First, *why* would you use LaTeX instead of e.g. Microsoft Word. There are a few good reasons to do so, especially when the document is getting larger than ten pages. LaTeX is a programming language and only updates when you say so. This results in working along without being bothered by a slow computer. Also figures and tables are handled more easily and LaTeX even helps you with the placement of them. Furthermore, equations can be written very clear and get a very clean look. This also holds for the cross-references, which are easily handled by just labelling the chapters, figures, tables, equations, etc. Last but not least, the handling of your citations is in a very structured and clear manner using LaTeX.

Next, *how* can you use LaTeX. There are two main options for using LaTeX: (1) online; and (2) offline. If your report is not too large and does not contain a lot of large pictures, you can use online LaTeX-compilers: e.g. Overleaf.[2] When you are writing a large document – like your thesis – it is recommended to work offline. Otherwise you have to pay because the time to compile will be too long for the free version. Offline editors can easily be found on the Internet; e.g. TeXmaker, TeXstudio, and many more.[3]

As briefly introduced, LaTeX is a programming language and to see a result – the document – you need to compile, or run, your LaTeX-code. In the online versions, this is easily done via the 'Recompile'-button. For offline editors, you will need to run through the code more than once to get all the (cross-) references right. The standard procedure for the offline editors is:

1. Run `pdflatex`
2. Run `bibtex`
3. Run `pdflatex`
4. Run `pdflatex`

For now, we will focus on the basics of LaTeX and some simple advises will be given to make life with LaTeX easier. We start with creating the main-file, which consists of a so-called preamble (Section 2.1), but is also the location in which all the other LaTeX-files are loaded; e.g. the title page (Section 2.2) and all chapters and/or sections of your document.

In LaTeX, typing the text is the same as in every other word-editor programme. Therefore, this part is not discussed in this document. What is elaborated on are the features that are slightly different; namely headings (Section 3), figures (Section 4), tables (Section 5), equations (Section 6), cross-references (Section 7), and references (Section 8).

2 Main-file

The main-file of your report will consist of a couple of parts. First of all, the preamble. This part is different and completely new compared to e.g. Microsoft Word, where such a thing does not exist. Nevertheless, this is a very important part in LaTeX at will therefore be elaborated on first, in Section 2.1.

Next, some nice features of LaTeX are presented, which are the title page in Section 2.2; and the table of contents in Section 2.3. Both easily automatically generated, but adaptable if wanted. This section will close with advice on the structure in Section 2.4 to keep your LaTeX-code clear and structured. This mainly focuses on a mapping structure, and how to use this in LaTeX.

[1] This document is written in the `article`-class.

[2] `www.overleaf.com`.

[3] A list of LaTeX editors can be found here: `beebom.com/best-latex-editors/`.

2.1 Preamble

The preamble is the location in your LaTeX-code where packages are loaded, the title is set, etc. LaTeX makes use of packages to minimise the computation time. Therefore, only load packages when needed. Otherwise, compiling your LaTeX-code will take longer, when not needed; just as with programming-codes like `matlab`. These packages are loaded by using the command:

```
\usepackage[]{}
```

The package itself is stated between curly brackets; and specifications and extra commands are typed between the square brackets, which are often left out.

When creating a new file in Overleaf, automatically a basis preamble is created, which you have to do by yourself when working offline. This preamble should always start with the document class,[4] where the report and article are the most interesting for students. The basic way to start a report is the following:

```
\documentclass[a4paper,twoside,openright]{report}
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\begin{document}
...
\end{document}
```

This report has some specifications between brackets: `[a4paper,twoside,openright]`. This means that the report is on an A4-paper; it is in the form of a booklet, so LaTeX makes the distinction between the left and the right paper; and a new chapter always starts at the right paper. Try this for yourself and see what it looks like. There are many other specifications and if you are looking for something, again, the Internet knows the answer.

The packages `inputenc` and `babel` are included for the writing; the `inputenc`-package make it possible to make use of the input of accented characters and `utf8` is a way to go. A hint on this topic: just accept this and move on; the English – as well as the Dutch – language does not make use of many (strange) accents, so it is not *that* important. The `babel`-package is nice to use for auto-correct. Without the package, LaTeX still gives some suggestions, but with the package loaded, its dictionary expends.

All the text before `\begin{document}`, is not directly included in the report. This is exactly where all the packages are loaded, but also other type-settings can be done; e.g. defining your own lay-out. The area between `\begin{document}` and `\end{document}` is all written into the PDF-file.

The most important thing you need to know about the preamble is the specification of the document class; importing the packages you need; and starting your document.

2.2 Title page

LaTeX offers the opportunity to create your own title page, but you can also make use of the standardised title page of LaTeX. Creating your own title page can be done specifying its lay-out in its own environment; so between:

```
\begin{titlepage}
...
\end{titlepage}
```

The standard title page of LaTeX can easily be used by typing `\maketitle`. Both commands must be placed within the document environment: thus between `\begin{document}` and `\end{document}`. Again, more information on the possibilities of making your own title page can be found on the Internet.

[4] Examples of classes can be found here: `tex.stackexchange.com/questions/782/what-are-the-available-documentclass-types-and-their-uses`.

2.3 Table of contents

Another, often used item in a report is the table of contents. This is as easily added as a standardised title page, by placing the following command inside the document environment: `\tableofcontents`. The 'depth' to which level the table of contents is presented can be regulated by you. Otherwise, LaTeX shows all numbered levels of headings (see Section 3).

2.4 Structure

It is advisable to put your preamble in your main-file and load the chapters/sections into this file. In this way, you can easily navigate through your report. The same holds for figures, `matlab`-files, etc. It is possible to save all your chapters in one map, your figures in another, andsoon. This will help you to keep track.

Loading of the chapters can be done via e.g. the command `\input{..}`. As an example, we have made a separate file for the introduction, which is a chapter, called `Introduction.tex`. This file is placed inside a map, called `Content`. The file is loaded – and so included in the document – as follows:

```
\input{Content/Introduction.tex}
```

The `\input{..}` command must be placed inside the document environment, whereas it is part of the content of your report. As you can see, first the mapping structure is typed in the command, separated by a slash: `/`. At the end, the LaTeX-file is typed. In many editors, the path and/or the file are suggested; to make it easier for you.

Lastly, LaTeX – just as is common in programming languages – has the option to include comments inside your text, which are not compiled as such. This is done by type-setting them behind the *percentage*-sign: %. This 'comment-mode' holds until you start a new line, but can be started half-way your text.

3 Heading

In LaTeX, headings are inserted via a command as given in Table 1. LaTeX automatically numbers the headings, except for the paragraph and the sub-paragraph, which are unnumbered. If you want an unnumbered heading, add an asterisk to the command: e.g. `\chapter*{..}` gives an unnumbered chapter-heading. The labels in Table 1 (last column) are for the cross-reference, which is elaborated on in Section 7.

Table 1: Available headings in LaTeX and their commands.

Heading	Code	Label
Part	`\part{..}`	`\label{part:..}`
Chapter	`\chapter{..}`	`\label{ch:..}`
Section	`\section{..}`	`\label{sec:..}`
Subsection	`\subsection{..}`	`\label{sec:..}`
Sub-subsection	`\subsubsection{..}`	`\label{sec:..}`
Paragraph	`\paragraph{..}`	n/a
Sub-paragraph	`\subparagraph{..}`	n/a

Unnumbered headings

Adding a label to an unnumbered heading does not work, whereas the cross-reference makes use of the numbers to refer to; as you can see at the end of the paragraph, referencing to Section 7. When you want to refer to an unnumbered heading, LaTeX refers to the closest heading in which the wanted one is situated.

4 Figure

To insert a figure in your LaTeX document, you first have to save (or upload) the figure to a location close to the LaTeX-files for convenience; e.g. in a map called `Figures`. Next, you have to load a few packages for controlling and placing the figure: (1) `graphicx`; and (2) `caption`.[5]

The nice thing about LaTeX is that it helps you to find the best location for your figures (as well as tables). In this, you have the option to give a preferred location of the figure; leave it completely to LaTeX to decide; or 'force' LaTeX to place the figure at the location you demand. This can be done with the `float specifier`, which has a couple of options: `t` for top of the page; `b` for bottom of the page; and `h` for here in the text. The addition of an exclamation mark (!) forces LaTeX to place the figure at that place in the text. It is also possible to put more than one specifier. LaTeX will than place the figure at the best location in which it uses the specifiers in the indicated order: e.g. `[ht]` will result in the placement of the figure here in the text, but if not possible at the top of the page.

To make it all clear and visible, an example of the placement of the figure, which is shown in Figure 1:

```
\begin{figure}[ht]
    \centering
    \includegraphics[height=3cm]{Figures/Figure1.png}
    \caption{Example of a figure}
    \label{fig:blue_man}
\end{figure}
```


Figure 1: Example of a figure.

As you can see, the figure is placed in the text at the location of the command, due to the specifier used: `[ht]`. The next line indicates that the figure should be centred after which the figure is loaded via the command `\includegraphics[]{}`. This command has an option to specify its size between square brackets; and the file is loaded using the curly brackets.

The size can be specified with the following commands: (1) width; (2) height; and (3) scale. The width and height are given as an absolute value or a relative one; e.g. compared to the text width. The scale is a multiplication factor of the original figure.

Sub-figures

Sub-figures are easy to make, and the code is very much a like a figure; even better said, it is placed inside the code for a figure. To do so, one more package needs to be loaded: the `subcaption`-package.

In the `figure`-environment, a `subfigure`-environment is created, including caption and label. The placement of sub-figures to each other is regulated via specifying the width of the `subfigure`-environment. Again, an example to make it all clear:

```
\begin{figure}[ht]
```

[5] The caption-package is actually solely needed for the caption of the gure. Because in a decent report (or article) every gure is labelled and has a caption, this package is already mentioned here.

```
    \centering
    \caption{Example of two sub-figures}
    \label{fig:subfig}
    \begin{subfigure}{.5\textwidth}
        \includegraphics[height=3cm]{Figures/Figure1.png}
        \caption{The first sub-figure, on the left side}
        \label{fig:sub_a}
    \end{subfigure}%
    \begin{subfigure}{.5\textwidth}
        \includegraphics[height=3cm]{Figures/Figure1.png}
        \caption{The second sub-figure, on the right side}
        \label{fig:sub_b}
    \end{subfigure}
\end{figure}
```

The result of the above code is shown in Figure 2 in which both sub-figures can be referred to separately: Figures 2a and 2b. For more information, make use of the wonders of the Internet.

Figure 2: Example of two sub-figures.

a: The first sub-figure, on the left side.

b: The second sub-figure, on the right side.

Note the fact that in this example the caption of the figure as whole (thus Figure 2) is placed above. One can choose to place it underneath or above depending on the location of `\caption{Example of two sub-figures}` and `\label{fig:subfig}` before or after the `subfigure`-environments. As long as the two commands are after each other – in which the `\caption{}`-command should lead the `\label{}`-command – it will work just fine.

TikZ-picture

There is also a function in LaTeX with which you can 'draw' using code, which is called Ti*k*Z and loaded using the `tikz`-package. The benefit can be a clear, simple sketch in the same formatting as your document. Figure 3 is an example of two functions plotted using Ti*k*Z: $y = \quad 0{:}05x^2 + 1{:}5$; and $y = 1 + 0{:}8 \sin x$. The code for this figure is:

```
\begin{figure}
    \centering
    \begin{tikzpicture}
        \draw[help lines,<->] (0,2) |- (5,0);
        \draw[domain=.1:4.9] plot (\x, {1.5-0.05*\x*\x});
        \draw[domain=.1:4.9,dashed] plot (\x,{1+.8*sin(\x r)});
        \node[below left] at (0,2) {$y$};
        \node[below left] at (5,0) {$x$};
    \end{tikzpicture}
    \caption{Example of two functions in Ti\textit{k}Z}
```

```
        \label{fig:TikZ}
    \end{figure}
```

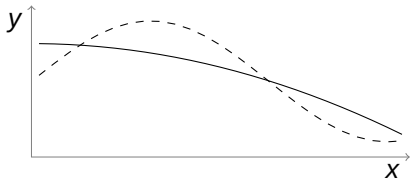

Figure 3: Example of two functions in Ti*k*Z.

A figure created in the `tikzpicture`-environment does not need to be placed inside the `figure`-environment, but can also be placed in the middle of the text. When the command-lines for the drawing become too much, one can also make use of the `\input{}`-command. Note the fact that `\includegraphics[]{}` is not used anymore.

Because of the widespread possibilities of this option, and the rather complex codes that can be involved for this topic, you are referred to Crémer (2011); who wrote *A very minimal introduction to TikZ*.

5 Table

The code for a table in LaTeX is rather complex and not that clear, especially for larger tables it can become a burden to see what is going on in the LaTeX-code. Fortunately, there is hope, which is – again – the Internet: on the Internet, there are many table generators in which you can clearly see what you are doing and produce a code for a table for you.[6]

Nevertheless, it is still good to know what is going on. The `table`-environment has many options – as with many things in LaTeX – but we will stick to the basics. For all extra's, use the Internet and experiment for yourself.

First things first, the standard code for a table, which is shown in Table 2 and includes a horizontal and a vertical line:

```
\begin{table}[ht]
    \centering
    \caption{An example of a table}
    \label{tab:ABCD}
    \begin{tabular}{c|cc}
        A & B & C \\\hline
        D & E & F \\
        G & H & I
    \end{tabular}
\end{table}
```

Table 2: An example of a table.

A	B	C
D	E	F
G	H	I

You can probably recognise the location specifier and the centering, just as with the `figure`-environment (Section 4). Also the caption and the label are the same as with a figure, but placed

[6] For example, `https://www.tablesgenerator.com`.

at the beginning so the caption is above the table instead of below.[7]

Next, the table is constructed in the `tabular`-environment, which starts by specifying the amount of columns, the text alignment, and the vertical lines. In our example, we keep it simple and we only use three columns, all text is aligned in the centre, and there is only one vertical line. Within the `tabular`-environment, the table is filled with content in which the &-sign is used as 'tab', and a new line is coded as \\. In this example, an horizontal line is added as well after the first row: `\hline`.

The `tabular`-environment can also be used without the `table`-environment. Nevertheless, without the `table`-environment, the centring and adding a caption and label is more of a burden and not well linked to the table itself (i.e. the `tabular`-environment). Therefore, when you want to make a table – as Table 2 – place all inside the `table`-environment; when you want to make an inline table-like object, feel free to only use the `tabular`-environment.

Table options

Because there are many options to modify your table to your needs – as already mentioned – these are not treated in this document. Nevertheless, a couple of examples of the possibilities are listed, to give you an idea what you can search for. A few examples are:

- Merge columns and/or rows;
- Rotate tables;
- Add footnotes to tables;[a]
- Tables covering more than one page;
- Sub-tables;
- Multi-line text inside one cell;
- Align numbers around the decimal point inside the table;
- Nice formatting of tables (search for *booktabs*).

[a] Adding a footnote to a table is not working, just as within the caption of a figure; i.e. footnotes cannot be placed within the `table`- and `figure`-environments, or any other *float*. Fortunately, there are a couple of ways to move around this.

6 Equation

One of the main pros of LaTeX is its clear, keen formatting of equations. There is no equation strange, complex, and big enough for LaTeX to be generated nicely. Take for example Equations (1) and (2), presenting the probability density function of the Weibull distribution, and the significant wave height of wind-waves, respectively.

$$f(x|\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left(-\left(\frac{x}{\lambda}\right)^{k}\right) & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{1}$$

[7] The automatic suggestion by LaTeX places the caption and label at the bottom of the `table`-environment { just as with the `figure`-environment. If you leave it there, the caption will be placed below the table. Nevertheless, in scientific writing the caption of tables is commonly above the table instead of below.

$$\frac{gH_s}{u_w^2} = 0{:}283 \tanh\left[0{:}578\left(\frac{gh}{u_w^2}\right)^{0{:}75}\right] \tanh\left[\frac{0{:}0125\left(\frac{gF}{u_w^2}\right)^{0{:}42}}{\tanh\left[0{:}578\left(\frac{gh}{u_w^2}\right)^{0{:}75}\right]}\right] \tag{2}$$

For the explanation of the `math`-environment, we will use easier formulae for clearance. To start with, there are three different formats: (1) inline; (2) outline, or unnumbered; and (3) numbered.

The inline format is the easiest and perfect for writing the symbols of parameters or small equations. You simply put the symbol or equation between dollar-signs: $F = ma$ is coded as `$F=ma$`. As you can see, an easy way to quickly write a small equation, which is always unnumbered. Because it has the same formatting as in equations, this method can also nicely be used to indicate the definitions of the parameters used in the equation.

The outline – or unnumbered – format is used for equations that are too long to place inline and are not referred to, and can be generated in two ways: (1) between double dollar-signs; or (2) with a starred `equation`-environment. The first one is comparable to the inline format: `$$F=ma$$` gives

$$F = ma$$

Even when placed in the middle of a paragraph; which was actually done in this case. This can make it sometimes slightly unclear what you are doing, when looking at the code. Therefore, the second option is more frequently used:

$$F = ma$$

This is coded as a 'normal' equation, but with an asterisk so it becomes unnumbered; just as with headings (see Section 3, but also the box Unnumbered headings). As you can see, in the compiled file there is no difference. Its code is:

```
\begin{equation*}
    F=ma
\end{equation*}
```

The numbered format is used for important equations to which you want to refer to in the text. The code is almost the same as for the unnumbered equation, but no asterisk (so it *is* numbered) and a label to refer to.[8] Equation (3) is coded as follows:

```
\begin{equation}
    F=ma
    \label{eq:Newton}
\end{equation}
```

$$F = ma \tag{3}$$

To get the most out of your equations, packages must be loaded in which certain features are stored. A good package to start with is the `mathtools`-package; e.g. without this package, Equation (1) could not be made. Also, the unnumbered equation can only be written using the double dollar-signs; and the option with the asterisk is not possible. Therefore, it is recommended to include this package as part of your default packages.

Sub-equations

Just as with figures and tables, you can also make sub-equations. To do so, the `mathtools`-package will work just fine. The big difference between the sub-figures and the sub-equations is in the coding; for sub-equations, you *start* with creating the `subequations`-environment and *then* create the `equation`-environment, which is opposite for the sub-figures (see the box `Sub-figures` in Section 4). Equations (4a) and (4b) are written with the following code:

[8] The numbered equation also works without label, but in that case you cannot refer to it.

```
\begin{subequations}
    \begin{equation}
        \frac{1}{\rho_{0}}\frac{\partial p}{\partial x} = fv
        \label{eq:x_dir}
    \end{equation}
    \begin{equation}
        \frac{1}{\rho_{0}}\frac{\partial p}{\partial y} = -fu
        \label{eq:y_dir}
    \end{equation}
\end{subequations}
```

$$\frac{1}{\rho_0}\frac{\partial p}{\partial x} = fv \tag{4a}$$

$$\frac{1}{\rho_0}\frac{\partial p}{\partial y} = -fu \tag{4b}$$

As you can see, the equations are automatically placed underneath each other. Two sub-equations as short as Equations (4a) and (4b) can easily be placed next to each other. To do so, you have to make use of so-called 'mini-pages'; the following code gives Equations (4a) and (4b) but placed next to each other, as shown in Equations (5a) and (5b):

```
\begin{subequations}
\begin{minipage}{.5\textwidth}
    \begin{equation}
        \frac{1}{\rho_{0}}\frac{\partial p}{\partial x} = fv
        \label{eq:x_dir}
    \end{equation}
\end{minipage}%
\begin{minipage}{.5\textwidth}
    \begin{equation}
        \frac{1}{\rho_{0}}\frac{\partial p}{\partial y} = -fu
        \label{eq:y_dir}
    \end{equation}
\end{minipage}
\end{subequations}
```

$$\frac{1}{\rho_0}\frac{\partial p}{\partial x} = fv \tag{5a}$$

$$\frac{1}{\rho_0}\frac{\partial p}{\partial y} = -fu \tag{5b}$$

7 Cross-reference

The cross-reference can become a real burden in many editors, but with LaTeX this is very easy and the numbering is always correct. There is only one thing to do, which is labelling. In Table 1, the label specifications of the headings are already listed, and the ones for figures, tables, and equations are already presented in Sections 4 to 6, respectively.

If you want to refer to one of these, you just use the command `\ref{label}` in which you replace 'label' with the label of your choice. This reference-method is the default of LaTeX. There have been made some other methods for referencing; e.g. the one highlighted below.

Automatic labelling

Where the default-setting for the cross-reference does only print the numbering of the labelled item, there are also packages that include the kind of the object referred to; e.g. Figure 1 instead of 1, which is a reference to the same figure.

One of these packages is the `cleveref`-package. This package allows – beside the inclusion of the object – to refer to more than one item; e.g. Sections 4 to 6.

This package uses the command `\cref{}` instead of `\ref{}` in which the same labels must be typed. The default of the package is to abbreviate the object's name and make a difference between `\cref{}` and `\Cref{}`, where the latter results in a capital letter as 'Eq. 3' and the first gives 'eq. 3'.

Referring to more than one object, is done by separating the different labels with comma's; e.g. `\cref{fig:sub_a,fig:sub_b}` gives fig. 2a and 2b. The objects separated by the comma – or comma's – do not need to be the same kind; figures and tables can be placed in the same `\cref{}`-command, and LaTeX will make something nice out of it.

The settings used in this document – which makes use of this package – are no abbreviations and capitalised labels: `\usepackage[noabbrev,capitalise]{cleveref}`. You are of course free to choose your own settings for the cross-references in your document, this package is solely an advice.

8 References

To make use of references/citations in your document, you need to make a separate file in which all the citations are listed. This file is called the `.bib`-file and should end with `.bib` instead of `.tex`; LaTeX will know that it is the bibliography-file and will read it as such. In this file, you have to place the details concerning the literature you want to cite. Once you cite e.g. an article, it is automatically added to the bibliography. Depending on the bibliography style you choose yourself, LaTeX adds a number of presents the name(s) and the year of publishing.

The bibliography can be loaded via various packages, one of them is the `natbib`-package in which different bibliography styles can be used; most common is the `apa`-style, which is also used in this document. The style is defined by the `\bibliographystyle{}`-command, which should be placed after the `natbib`-package is loaded. With these commands, no bibliography is printed yet, because you have to state where LaTeX should place it. This is done by `\bibliography{file_name}` in which `file_name.bib` is the bibliography-file.

This file is contains all the information on the literature you want to cite. This literature is placed in this file following a couple of statements; depending on the kind of literature, certain input parameters are essential.[9] As an example, the code for an article is presented below:

```
@article{GCP2010,
    author      =   {G\'{o}mez, E. A. and Cuadrado, D. G. and Pierini, J. O.},
    title       =   {Sand transport on an estuarine submarine dune field},
    journal     =   {Geomorphology},
    volume      =   {121},
    pages       =   {257--265},
    year        =   {2010}
}
```

After the at (`@`), the kind of literature is typed, and you start by giving a name to the citation; give a name that is not too long, whereas this is kind of the 'label' of the citation to which is referred in the document. Important is the separation of the different statements by comma's. Also, the authors are separated with '`and`'. When the last name is typed before the first name(s) or initials,

[9] For a list of the possible literature and the statements for the bibliography- le, have a look at e.g. https://nl.overleaf.com/learn/latex/Bibliography_management_with_bibtex.

a comma should be placed after the last name; `G\'{o}mez, E. A.` instead of `G\'{o}mez E. A.`. If the first name(s) or initial(s) are placed before the last name, this comma is not needed; thus `E. A. G\'{o}mez` instead of `E. A., G\'{o}mez`.

The above article gives the following citation in the text – Gómez et al. (2010) – and can be found in the references. As you can see, LaTeX replaces the authors automatically with *et al.* when there are more than two authors. This citation is placed in the text using the `\cite{GCP2010}`-command; as you can see, the 'label' of the citation is used to refer to it.

Citing features

To get more control on the presentation of your citations, the `cite`-package can be used. This package works the same as described above, but has an extra feature: you can make the distinction between 'Gómez et al. (2010)' and '(Gómez et al., 2010)'; the latter is the same, but the first enables you to cite and use the authors as part of the text. These two have the command `\citet{GCP2010}` and `\citep{GCP2010}`, respectively.

9 Exercise

The following exercises are there for you to check whether you get LaTeX or not, and hopefully you will encounter some errors, which you can ask (or use the mighty Internet) of course. This exercise will result in a document without much content, but just helps you to get to know LaTeX. If you want to put a report of yourself in LaTeX, you can also use this report as content and try to place everything in it as you wish. These exercises are just a way to get you started, whereas the best way of learning to work with LaTeX is just doing it. The exercises might contain some features that are not elaborated in this document; this is on purpose, to get you using the Internet to help you, whereas there are millions of options and possibilities that are not encountered in this document that you probably want to use – now or in the near future.

Keep in mind that their might be more than one solution to a 'problem' and you can choose the one that suits you best. And never forget, the Internet is your biggest friend.[10] Well, good luck!

9.1 Main-file

The exercises will be part of a report. First, set-up a basic preamble, including a nice title etc. from which you are going to work. Next, make separate LaTeX-files for every exercise, which are presented below; every part of the exercise is a new chapter and loaded in the main-file. In this way, you immediately have to construct a clear structure in your document. Also include a table of contents, so you know where to find all the exercises in your document.

Fill up the document with some text, which does not need to have a meaning; e.g. Lorem Ipsum[11] or `blindtext`.[12] Separate the text with sections, subsections, etc. to make a structure in the text as well as to create a full table of content.

When done, have a look at your table of content when you use a lot of heading-levels and find a way to minimise the 'depth of your table of content'. Furthermore, if you have time, you can even try to make your own title page; play around with the features of LaTeX in which you probably want to include some spacing; horizontally and/or vertically.

Hint: Do not forget to load the packages needed for the exercises, as presented in the sections above. All listed packages are recommended to become part of your default packages; the packages you automatically load, because you know you will use them.

[10] If you search for a `solution' to your `problem', it is recommended to look if `tex.stackexchange.com` is among the suggestions. This is the StackOver ow for LaTeX and commonly gives very clear answers to your questions in which an explanation is commonly included.

[11] Can be imported in LaTeX by copying it from the Internet, or loaded via the package `lipsum`.

[12] Can be loaded via the package `blindtext`, and has a couple of funny features; see the Internet.

9.2 Figure

Place a figure of your own choice – e.g. downloaded from the Internet – at the bottom of the page, and aligned on the left side of the paper. Include a nice caption and make sure the figure is not too big; e.g. give it a height of four centimetres, or a width of 30% of the text width.

9.3 Table

Reproduce Table 3 and try other lay-outs and/or contents to your own preferences. First, give it a try to make the table without a table generator; as mentioned in Section 5. In this way, you will get a better feeling what is going on.

Table 3: Flood barriers in the Netherlands as part of the Delta Works; start and finish of construction.

Flood barrier	Beginning	Inauguration
Stormvloedkering Hollandse IJssel	1954	1958
Haringvlietdam	1958	1971
Oosterscheldekering	1960	1986
Maeslantkering	1988	1997
Hartelkering	1991	1997

If you really want to know how to make tables, try to duplicate Table 4 without a table-generator. Table 4 is especially interesting concerning the lines in it, and thus is pretty ugly so you encounter most of the features possible.

Table 4: Overall results.

Treatment	Females					Males				
	V1	V2	V3	V4	V5	V1	V2	V3	V4	V5
Placebo	0.5	0.5	0.5	0	0.5	0	0.5	0.5	0.5	0
Non-placebo	1	1	1	1	0.5	1	1	0.5	0.5	0.5

Hint: For Table 4 you need the packages `multicol` and `multirow`.

9.4 Equation

Try to duplicate Equation (6), which is clearly slightly more difficult than the ones presented in Section 6. This equation has some interesting possibilities of equations with LaTeX. If you are up for a challenge, try to duplicate Equations (1) and (2) as well. Do not forget to label the equation(s), which is needed for the next exercise (Section 9.5).

$$\int_{h_S}^{0} \frac{\partial p}{\partial x} \mathrm{d}z = \int_{h_S}^{0} f v + \frac{\partial_{s;x}}{\partial z} \quad \mathrm{d}z \tag{6}$$

9.5 Cross-reference

In the above exercises – Sections 9.2 to 9.4 – you have created a couple of objects to which you can refer to. In this exercise, refer to all these objects and maybe even try to come up with some interesting text around the cross-references. Also include the chapters and/or sections you have created around the exercises.

If you are making use of the `cleveref`-package, try to combine your figure from Section 9.2 and your table from Section 9.3 in one `\cref{}`-command to see how LaTeX deals with it.

9.6 References

Create a bibliography-file in which you cite the article by Martin and Fitzwater (1988); the details can be found in the list of references below. Have a look at the code for Gómez et al. (2010) in Section 8 if you do not know what the different fields mean.

When you are up for a real challenge concerning citations, try to reproduce the article by Piccolo et al. (2008).

Hint: Piccolo et al. (2008) is part of a collection of articles presented in the book *Perspectives on Integrated Coastal Zone Management in South America*. Use the Internet to find out how to cite such a source.

References

Crémer, J. (2011). A very minimal introduction to Ti*k*Z. `https://cremeronline.com/LaTeX/minimaltikz.pdf`.

Gómez, E. A., Cuadrado, D. G., and Pierini, J. O. (2010). Sand transport on an estuarine submarine dune field. *Geomorphology*, 121:257–265.

Martin, J. H. and Fitzwater, S. E. (1988). Iron deficiency limits phytoplankton growth in the north-east Pacific subarctic. *Nature*, 331:341–343.

Piccolo, M. C., Perillo, G. M. E., and Melo, W. D. (2008). The Bahía Blanca estuary: an integrated overview of its geomorphology and dynamics. In Neves, R., Baretta, J. W., and Mateus, M., editors, *Perspectives on Integrated Coastal Zone Management in South America*, pages 219–229. IST Press.