LaTeX-workshop (Instructions)

De Leidsche Flesch

October 6, 2016

Contents

1 Introduction

This is the manual for the LaTeXworkshop of De Leidsche Flesch. Please read through this manual first. Then you can start working on the corresponding exercises. When working on these exercises, it is intended that you use this manual and the internet to look things up.

You will probably stumble upon several errors, because you made a mistake *somewhere in the code*. This is the intended design of the workshop, because later, when using LaTeX , you will definitely make more mistakes, and therefore it is important that you are able to solve these problems. Furthermore, please remember that there are often several ways to portray things, with a slightly different layout. This is simply a matter of preference.

If you find making LaTeXdocuments quite difficult, please utilize the online help. The wikibook about LaTeX(`https://en.wikibooks.org/wiki/LaTeX`) is very approachable and most occuring situations are extensively described in it. If you can not find the command of a certain symbol, please try `http://detexify.kirelabs.org`. On this website, you can draw the symbol you are looking for, and it will find the corresponding command for you.

1.1 Beginning

To start with the workshop, you will have to follow the following steps:

1. Start your computer and log in.

2. Click on the startmenu → all programs → MiKTeX 2.9 → TeXworks [1]

3. Read this manual. The manual and exercises can (also) be found on our website[2].

4. Take your bundle of exercises.

5. You can now start making the exercises. For inspiration, read the manual carefully and scour the internet.

An online TeX editor exists for a few years: Sharelatex[3], which is for making LaTeXdocuments *in the cloud*. It works like Google Docs: it is accessible from every computer and all files you make will be saved online. This way, you can never forget your homework.

2 New document

A LaTeX document is made out of 2 parts. The first part is the preamble. The things written here will be valid throughout the whole document, such as the right packages and fonts. The second part starts at `\begin{document}` and ends at `\end{document}` The contents of your document are inbetween these two commands.

You can put comments in your .tex file, without them being displayed in your PDF. You can accomplish this by using `%` in front of your commentary.

---

[1]You can download the MiKTeX program at home at `http://miktex.org/download`, this is an easy program for making LaTeX code.

[2]`http://www.deleidscheflesch.nl/p/latex-workshop`
After the workshop, these files will stay online.

[3]`https://www.sharelatex.com`

2.1 Preamble

1. At the top, you put `\documentclass{article}` This command tells your compiler what kind of text you are writing, in this case an article. It specifies for example the font size. The article class is the best in almost every situation.

2. Directly below this you put the packeges you want to use. These make it it possible to add extra functies to LaTeX that aren't standard. Add a package by putting `\usepackage{packagenaam}` at the top of your document.

   A couple of commands that are often used in documents are:

   1) `\usepackage[a4paper]{geometry}` It specifies the dimensions of your document to the A4 format.

   2) `\usepackage[dutch, english]{babel}` This one is for correct translations and word wrapping. For example 'Chapter' instead of 'Hoofdstuk'. Here also Dutch is loaded, so you could use Dutch quotes for example. The last language in the block is the main language of the document.

   3) `\usepackage{parskip}` For a correct display of paragraphs.

   4) `\usepackage{amsmath, amssymb}` A package that displays formulas and other mathematical object in a better looking way.

   5) `\usepackage{graphicx}` To add images to your document.

   6) `\usepackage{enumerate}` To easily make numbered lists.

   7) `\usepackage{url}` To display urls properly.

   8) `\usepackage{float}` To ensure proper placement of images in the document

   9) `\usepackage[hidelinks]{hyperref}` To make a clickable table of contents, where clicks transport you to that part of the document.

   10) `\usepackage{color}` To type text in color.

3. Put the following commands in your peamble:

   `\title{titel}`
   `\author{auteur}`
   `\date{\today}`, the command `\today` automatically adds the correct date.

4. the last thing you do is to add `\begin{document}` and `\end{document}` Between these two commands you put your text.

5. Compile your document.

2.2 Text in a document

2.2.1 Title and Author

At the top of your document put `\maketitle` This makes sure the information you put in the preamble as the title will be at the right place in the document. In this case a title, author and date.

2.2.2   Table of Contents

You make a table of contents with the command `\tableofcontents` Then you will want to start a new page using `\newpage`

2.2.3   Headers

An Article document in LaTeX uses sections, subsections, subsubsections and paragraphs. 'New document' in this document is a section for example. You make these by typing `\section{Name header}`

The header 'Text in document' is a subsection. You make these by typing `\subsection{Name Header}`

the header 'Headers' is een subsubsection. You make these by typing `\subsubsection{Name Header}`

Paragraph   Then you also have paragraph headers, you make these by typing `\paragraph{Naam paragraaf}` This header is a paragraph type.

2.2.4   Paragraph

You can make paragraphs by leaving an empty line between the text.

2.2.5   Enumerations and lists

You can make a non-numbered enumaration using `\begin{itemize}`. A dot in your enumaration is then made using `\item` Behind this command you type a spacebar and then the tekst you want behind the dot. When your enumaration is done you close it using `\end{itemize}`

- This is the first item of a non-numbered enumaration
- This is the second
- This is the third

A numbered enumaration, as seen earlier in this guide, is made using `\begin{enumerate}` You create numbers by again using `\item` te typen. Again you close of with `\end{enumerate}` This will give you:

1. Example 1
2. Example 2

If you want a differenf font or lay-out, you type `\begin{enumerate}[a.]`. then you'll get:

a Example 1

b Example 2

In the same way you can enumarate using roman numerals using `\begin{enumerate}[i.]`

  i. Example 1

 ii. Example 2

3 Text

Just like Microsoft Word, LaTeX provides the option of presenting text in different styles. A few of the possibilities are listed below.

- **bold**, type `\textbf{text}`

- *italics*, type `\textit{text}`

- color, typ `\textcolor{color}{text}` or `{\color{color}text}`.

- small, type `\small` followed by text.

- big, type `\large`

- bigger, type `\Large`

- even bigger, type `\huge`

- biggest, type `\Huge` .

Type `\normalsize` after this to continue in the usual textsize.

4 Math Mode

4.1 Display Equations

If you want formula's and equations in your reports then you will have to work in "Math Mode". There are multiple way to enter Math Mode, the 3 most used methods are *math*, *displaymath* and *align*, in which you can type mathematical formula's. In *math* the mathematical symbols will just appear in the same line[4], like so: $a^n + b^n = c^n$. This environment can be invoked in multiple ways. The most used way is putting `$`-signs around the equations.

The *displaymath* environment is used to center on the mathematical symbols, like:

$$\frac{\hbar^2}{2m}\nabla^2\Psi + V(\mathbf{r})\Psi = -i\hbar\frac{\partial\Psi}{\partial t}.$$

This environment can be invoked by putting the equation between the following brackets: `[...]`.

Lastly we have the *align* method, with this environment mathematical symbols will be centered en they will get a reference number behind the equation, like so:

$$\int_{-\infty}^{\infty}\frac{\cos x}{x^2+1} = \frac{\pi}{e} \tag{1}$$

$$\int_{-\infty}^{\infty}\frac{\cos x}{x^2+1} = \frac{\pi}{e} \tag{2}$$

---

[4]for some reason this does not always work for titles. You will need the package *fixltx2e* to make this work

To display an equation like this use `\begin{align}` and `\end{align}`. If you want to use this method without the reference numbers, type `\begin{align*}` and `\end{align*}`. To list multiple equations below eachother, type `\\` behind the equations. Use the `&`-sign to align the formula's correctly, most often you use the `&`-sign before the `=`-sign in the formula. If you are feeling adventurous, try finding out what happens if you use multiple `&`'s in the same line.

4.2 Superscript and subscript

When working in Math Mode, subscript (`_`) and superscript (`^`) are often used. This way we can for instance display $a_1, a_2, \dots, a_{n-1}$ with the following code: `a_1,a_2,\dots,a_{n-1}` and the equation $a^b \cdot a^c = a^{b+c}$ can be constructed by typing `a^b\cdot a^c=a^{b+c}`. Pay attention that if you want to subscript or superscript more than a single symbol you have to put { and } around it.

4.3 Roots and fractions

When using math you will often encounting roots and fractions. For displaying the fraction $\frac{a}{b}$ type the code: `\frac{a}{b}` and for displaying the root $\sqrt{a}$ gebruiken we `\sqrt{a}` and for $\sqrt[n]{a}$ we use `\sqrt[n]{a}`.

4.4 Speciale symbols

There are a lot of symbols that are used in math. In this paragraph a number of them will be named, but there are more. To quickly find which code belongs to which symbol you can use the website or the app Detexify[5]. Example of symbols are integrals, but also for instance α and β. Below are the most used symbols of the Greek alfabet:

	code		code		code		code		code		code
α	`\alpha`	β	`\beta`	γ	`\gamma`	δ	`\delta`	ϵ	`\epsilon`	ε	`\varepsilon`
ζ	`\zeta`	θ	`\theta`	λ	`\lambda`	μ	`\mu`	ξ	`\xi`	π	`\pi`
ρ	`\rho`	σ	`\sigma`	τ	`\tau`	ϕ	`\phi`	φ	`\varphi`	χ	`\chi`
ψ	`\psi`	Γ	`\Gamma`	Δ	`\Delta`	Λ	`\Lambda`	Φ	`\Phi`	Ψ	`\Psi`

Apart from symbols of the Greek alfabet, we also have standard functions, for which commands exist in LaTeX . These standard functions will be displayed when the appropiate command is used. The most used commands for functions are:

	code		code		code		code		code		code
arccos	`\arccos`	arcsin	`\arcsin`	arctan	`\arctan`	arg	`\arg`	sinh	`\sinh`	cos	`\cos`
cosh	`\cosh`	deg	`\deg`	det	`\det`	dim	`\dim`	tan	`\tan`	ker	`\ker`
lim	`\lim`	log	`\log`	max	`\max`	min	`\min`	sin	`\sin`		

Naast de tekens uit het Griekse alfabet en de standaard functies hebben we ook nog kwantoren. De kwantoren kunnen zich aanpassen naar de omgeving waarin zij zich bevinden. Zo worden

---

[5]`http://detexify.kirelabs.org/`

zij groter weergeven in *displaymath* dan in *math*. De onderstaande kwantoren worden het meest gebruikt:

	code		code		code		code
$\sum$	\sum	$\bigcap$	\bigcap	$\iint$	\iint	$\bigcup$	\bigcup
$\prod$	\prod	$\int$	\int	$\iiint$	\iiint	$\oint$	\oint

For $\sum$ and $\prod$ and lim it looks cleaner if in the *displaymath* environment the upper en lower limits are directly above or below the quantifier. You can use subscript and superscript to achieve this. For example you get:

$$\sum_{n=0}^{\infty} \frac{1}{n}$$

by using the next code: `\sum_{n=1}^{\infty} \frac{1}{n}`. You can make dubble integrals look cleaner by using the command `\limits`. This makes sure the limits will stand in the right place. For example compare:

$$\iint_D \text{ with } \iint\limits_D .$$

obtained by the code `\[\iint_^D \text{ with } \iint\limits_D\]`. The second one is the correct notation.

If you want to use $\sum$ or $\prod$ in the *math* environment and still want the upper en lower limits directly above or below the quantifier, you can use the command `\displaystyle`. So for example $\displaystyle\sum_{n=0}^{\infty} \frac{1}{n}$ is obtained by the code `\displaystyle\sum_{n=1}^{\infty} \frac{1}{n}`.

Finally there are a few symbols that are useful to know.

	code		code		code		code
$\pm$	\pm	$\times$	\times	$\setminus$	\setminus	$\cdot$	\cdot
$\leq$	\leq	$\in$	\in	$\subset$	\subset	$\subseteq$	\subseteq
$\cong$	\cong	$\geq$	\geq	$\approx$	\approx	$\mid$	\mid
$\equiv$	\equiv	$\sim$	\sim	$\neq$	\neq	$\leftarrow$	\leftarrow
$\Leftarrow$	\Leftarrow	$\leftrightarrow$	\leftrightarrow	$\Leftrightarrow$	\Leftrightarrow	$\mapsto$	\mapsto
$\hookleftarrow$	\hookleftarrow	$\rightarrow$	\rightarrow	$\Rightarrow$	\Rightarrow	$\updownarrow$	\updownarrow
$\Updownarrow$	\Updownarrow	$\uparrow$	\uparrow	$\Uparrow$	\Uparrow	$\downarrow$	\downarrow
$\Downarrow$	\Downarrow	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\forall$	\forall	$\exists$	\exists	∞	\infty	∇	\nabla
∂	\partial						

For comparative operators such as $\leq$, $\in$ and $\subset$ you can create the negavtive operator by putting `\not` for the operatot. Doing that you get $\nleq$, $\notin$ and $\not\subset$ for example.

4.5 Matrices

There are different types of matrices. The most frequently used matrix is the matrix enclosed in round brackets as delimiters. Another matrix you will probably use a lot is the matrix between straight lines as delimiters. You will use this for calculating the determinant. There are a lot of other types of matrices, but you will not use them that often.

		LaTeX-code
Ordinary	$\begin{matrix} a & b \\ c & d \end{matrix}$	`\begin{matrix} a & b \\ c & d \end{matrix}`
Between round brackets	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	`\begin{pmatrix} a & b \\ c & d \end{pmatrix}`
Between straight lines	$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	`\begin{vmatrix} a & b \\ c & d \end{vmatrix}`

5 Tables

There are different ways to create a table in a LaTeX-file. The most frequently used are the table
(`\begin{tabular}[position of the table]{layout of the table}`) and the array
(`\begin{array}[\textit{position of the table}]{layout of the table}`). You can only
create an array in a math evironment.
To separate the elements in a row, you use `&` and to close a row you use `\\`. To insert lines
between columns, you place `\hline` in the code the table was created with.

You can use the next commands to determine the position of the table:

Command	Result
`t`	De tabel is met zijn top gealigneerd op de onderkant van de regel.
`b`	De tabel is met zijn onderkant gealigneerd aan de onderkant van de regel.
`c`	De tabel is gecentreerd rond de onderkant van de regel.

Bij het bepalen van de opmaak, bepaal je ook het aantal kolommen waaruit de tabel bestaat. Om
te bepalen hoe deze kolommen eruit zien kunnen de volgende commando's worden aangeroepen:

Commando	Resultaat
`l`	Adds a column with left-justified text.
`r`	Adds a column with right-justified text.
`c`	Adds a column with centered text.
`p{x}`	De tekst wordt uitgevuld over een breedte `x`.
`*{x}{opmaak}`	Herhaald de structuur in `opmaak` `x` keer.
`|`	Adds vertical line.
`||`	Adds dubble vertical line.
`@{text}`	Gebruikt `text` als verticale lijn.

Example

Time in months	Number of rabbits	Computation
0	1	1
1	2	1+1
2	3	2+1
3	5	2+3
4	8	3+5
5	13	5+8
6	21	8+13
$\vdots$	$\vdots$	$\vdots$

This table can be invoked by the next code:

```
\hline
Time in months & Number of rabbits & Computation \\
\hline
0 & 1 & 1 \\
1 & 2 & 1+1  \\
2 & 3 & 2+1 \\
3 & 5 & 2+3 \\
4 & 8 & 3+5 \\
5 & 13 & 5+8 \\
6 & 21 & 8+13 \\
$\vdots$ & $\vdots$ & $\vdots$ \\
\hline
```

6   Inserting images

It is possible to include images to LaTeX. You first need to place `\usepackage{graphicx}` in your preamble. This package makes sure you can include images. Secondly you use the code `\includegraphics{picture.jpg}` to actually include the image. The image needs to be in the folder of your LaTeX-code to actually include it. Be aware: LaTeX can only include `.png`- and `.jpg`-images with the `graphicx` package.

You can use `\includegraphics[scale=...]{picture.jpg}` to alter the size of your picture. You can give the scale of the images, but you can also use `width=...\textwidth`. This way the images will also be scaled to the desired size.

6.1   Placement of the picture

It is often difficult to place the picture on the desired spot. Therefore you need to place the picture in a floating environment. This can be accomplished by

```
\begin{figure}[]
\centering
\caption{Caption of picture}
\includegraphics{width=0.5\textwidth]{picture.jpg}
\end{figure}
```

Commands can be given between the brackes of `\begin{figure}[]` for the placement of the picture. Below you can find some commands that are useful.

Command	Result
h	Place environment here.
t	Place environment on the top of the page.
b	Place environment on the bottom of the page.
!	Override LaTeX's internal parameters for determining "good" float positions.

As you can see in the code above, it is possible to include a caption to your image. This can be done by placing `\caption{Description}` in the floating environment of the image. The description will then be put under the image. It is also possible to place the description next to the image by the following code:

```
\begin{SCfigure}
\caption{Short description}
\includegraphics[width=0.5\textwidth]{image.jpg}
\end{SCfigure}
```

It is possible to wrap the text around the image. The image will be put in the text. First you need to include the package: `\usepackage{wrapfig}`. To place the image you need to use the following code:

```
\begin{wrapfigure}{r}{0.5\textwidth}
\includegraphics[width=0.45\textwidth]{picture.jpg}
\caption{Short description}
\end{wrapfigure}
```

The placement of the image can be determined by the text between the two braces with `r` in the example. Next there are a few nifty commands that are useful for placement:

commando	resultaat
r	At the right side.
l	At the left side.
R	At the right side in floating environment, to prevent splitting over pages.
L	At the left side in floating environment, to prevent splitting over pages.

To conclude it is also possible to put different pictures next to or above each other. This can be useful to compare graphs. You first need to include `\usepackage{subfig}` in the preamble. The following code can then be use to build a block of four images of two rows and two columns.

```
\begin{figure}
\centering
\subfloat[Description 1]{\includegraphics[width=0.3\textwidth]{graph1}}
\subfloat[Description 2]{\includegraphics[width=0.3\textwidth]{graph2}} \\
\subfloat[Description 3]{\includegraphics[width=0.3\textwidth]{graph3}}
\subfloat[Description 4]{\includegraphics[width=0.3\textwidth]{graph4}}
\caption{Graphs of ...}
\end{figure}
```

7 Impliment code

For implimenting code you can use the package *listings*. This gives an environment *lstlisting*. You put yout code between `\begin{lstlisting}` and `\end{lstlisting}`. However, you need to specify which programming language you use; If you use C++ for example put `\lstset{language=C++}` *above* `\begin{document}`.

De LaTeX-code

```
\begin{lstlisting}
  int main (int argc, char** argv) {
```

```
    std::cout << "Hello world" << std::endl;
    return 0;
  }
\end{lstlisting}
```

has for example

```
  int main (int argc , char∗∗ argv) {
    std :: cout << ”Hello ␣ world” << std :: endl ;
    return 0;
  }
```

as output.

You can easily impliment the file by using:

```
\lstinputlisting{bestandnaam.extensie}
```

8 References

When numbering things you can use the command `\label{HereAName}` The exact location for this code is troublesome; `\label{...}` remembers simply the last generated number in the current scope. When you have put `\label{...}` on the right spot, then you can use the number with `\ref{HierEenNaam}`. You can also use `\pageref{HierEenNaam}` to refer to the page where the label is placed.

If you want a footnote in your text use `\footnote{This is a footnote}`[6].

For extensive references use

```
\begin{thebibliography}{99}
    \bibitem{afbeeldingen} Wikibooks, \emph{LaTeX/Floats, Figures and Captions
        --- Wikibooks{,} The Free Textbook Project}.
        \url{http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions}
\end{thebibliography}
```

Afterwards, you can use `\cite{afbeeldingen}` to refer to the url with more information about the imgage [?].

For references it is important to compile the code twice. Otherwise there will appear questionmarks or old numberings.

9 Other interesting packages

Beside all this, there are numberous packages to make your life easier. You should search for them on the internet!

- `\usepackage{hyperref}` References within your document, url's and clickable tables of content in the `.pdf`. Look into the options of this package. Tables of contents can be created by using `\tableofcontents`.

---

[6]This is a footnote

- `\usepackage{beamer}` To create beamer-presentations (powerpoint).

- `\usepackage{fancyhdr}` For intelligent headers en footers on your page

- `latexmk` Is able to automise the whole compileproces.

- `\usepackage{sidecap}` For captions beside your floats instead of under or above them.

- Did you know, `\\`, the command for linebreak, has an optional argument. try for example `\\[5cm]` .

References

[1] `http://en.wikibooks.org/wiki/LaTeX`

[2] Wikibooks, *LaTeX/Floats, Figures and Captions — Wikibooks, The Free Textbook Project.* `http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions`

[3] `https://www.physics.leidenuniv.nl/bachelor/80-education/606-education-thesis-templates`