LaTeX and friends

Tools and technologies of (scientific) writing

Stefan Huber

ITS, FH Salzburg

March 16, 2022

Overview

- LaTeX
  - Philosophy and history
  - Basic document structure
  - Formatting
  - Multi-lingual
  - Math
  - References
  - Figures and tables
- Tooling
  - Distributions and compilers
  - Editors and tools
- Bibliography
  - biber
  - management software
- Creating figures
  - Ipe
  - TikZ
- Advanced topics
  - Presentations
  - Siunitx, blog article on units
  - Glossaries and acronyms
  - Index

Caveat

This material is opinionated at times. Since this is not a university course, I took the freedom.

History

- Donald Knuth, one of the fathers of computer science, was lacking a typesetting system for his opus magnum *The Art of Computer Programming*.
- He invented TEX and released it in 1978.
- Leslie Lamport release in 1984 LATEX, a macro ecosystem that is about content and structure rather than formatting (plain tex).
- Since then LATEX became the standard in academia, especially in mathematics, computer science, physics, chemistry, economics, engineering, parts of psychology and political science, et cetera. Let's say, everywhere where math and other formal notations play an important role.

Philosophy

LaTeX is not WYSIWYG:

- It is a markup language. You tell LaTeX what things are, not how to format them.
- A LaTeX compiler turns `.tex` files into PDF files, or other output formats.

```
Spaces    do not   count in   the
source. Note that end of sentence
has   a diff.\ spacing. See?

Also note -- like here -- that
there is an en-dash. English
people---none here?---also use
em-dash. And $4-3 = 1$.
```

Spaces do not count in the source. Note that end of sentence has a diff. spacing. See? Also note – like here – that there is an en-dash. English people—none here?—also use em-dash. And $4 - 3 = 1$.

Philosophy

I think there are two sources for the power of LaTeX:

- Its algorithms for top-notch quality in typesetting.
- Its ecosystem of packages that provide a rich set of macros for all kind of domains. Your own macros avoid repeating yourselves.

And since `.tex` files are source code, we have powerful tools:

- Your favorite text editor, e.g., with snippet completion.
- Git for versioning and collaboration.

Contrast to word processors

- The typesetting quality of Word is no match to LaTeX. InDesign is better but still no match: `http://www.rtznet.nl/zink/latex.php?lang=en`.
- The things you input is limited to the keys on the keyboard plus some (often annoying) semi-intelligent auto correction mechanisms.
- Starting the formula editor is laborious, so you don't, say, for each inline math expressions. And even then the formula editor is no match for the power of LaTeX.

Summary

Office software is for office business. Excel is no database and no statistics software, but a spreadsheet software. Word is no desktop publishing and no document preparation software with high-quality typesetting, but for office-type documents.

I have used LaTeX for:

- Books, publications, presentations, technical concepts, research project proposals, lecture notes, exams, lab assignments, letters, applications, guidelines.

Contrast to word processors

```
Newton says that a force $F$ applied
to a mass $m$ leads to acceleration
$a$ such that
\[ F = m \cdot a. \]

The gravitational field of the earth
causes an acceleration $g =
\SI{9.81}{\meter\per\square\second}$.
% Note that m (meter) ist not m (mass)
```

Newton says that a force F applied to a mass m leads to acceleration a such that

$$F = m \cdot a.$$

The gravitational field of the earth causes an acceleration $g = 9.81\,\mathrm{m\,s^{-2}}$.

Document class

- We need to tell what type of document we author: article, report, book, beamer, et cetera.
- We pass options, e.g., to set paper size. We tell some meta information using specific commands.

```
\documentclass{article}

\begin{document}
Hello world!
\end{document}
```

```
\documentclass[a4paper, 11pt]{book}

\title{Philosophiae Naturalis Principia
    ↪ Mathematica}
\author{Isaac Newton}
\date{5. Juli 1686}

\begin{document}
\maketitle % Prints title, author and date
\end{document}
```

- We call a begin-end block an environment.
- The content is in the document environment. Everything before is called the preamble.

Document structure

Different document classes provide different hierarchical structure elements.

- The stared versions, like `\section*`, are without numbering.
- The table of contents can be automatically generated.
- When generating a PDF, the PDF content structure is generated, with hyperlinks, if the `hyperref` package is included.

```
% A typical document content
\maketitle
\tableofcontents

\chapter{The first chapter}
\section{A section}
\subsection{A subsection}
\paragraph{A paragraph.}
```

	part	chapter	section	subsection	subsubsection	paragraph	subparagraph
book	■	■	■	■	■	■	■
report		■	■	■	■	■	■
article			■	■	■	■	■

Basic formatting

```
1  ``Say `hi!' to her''. Better
2  \enquote{enquote} from the csquotes
3  package \ldots
4
5  Instead of \underline{underline} or
6  \textbf{bold font}, use
7  \textit{italics} or \emph{emphasize}.
8
9  \LaTeX\ assumes periods mark an end of
10 sentence, except after an upper case
11 letter. We specify exceptions, as in
12 fig.~5 or THIS\@. See?
```

"Say 'hi!' to her''. Better "enquote" from the csquotes package ...
Instead of underline or **bold font**, use *italics* or *emphasize*.
LaTeX assumes periods mark an end of sentence, except after an upper case letter. We specify exceptions, as in fig. 5 or THIS. See?

Attention

Word users sometimes have a compulsion to force formatting details also in LaTeX. They typically make it worse. If you change the formatting, act with care. In general, less is more.

Font styles

- A font superfamily provides different styles, like bold, italics, typewritter, and so on.
  - If you change the font family then make sure you also choose a fitting math font
  - Well designed fonts are rare. The default LaTeX font by Knuth (Computer Modern) is excellent. Palantino with Euler math font is also excellent.
- You can change the base font size as documentclass option. Based on that different relative font sizes are available.
  - E.g., when designing a custom title page or developing your own document class.

```
\textrm{roman}, \textsf{sans serif},
\textsc{Small Caps},
\texttt{typewriter}, \textbf{bold},
\textit{italics}

\Huge Huge \huge huge \LARGE LARGE
\Large Large \large large
\normalsize normalsize \small small
\footnotesize footnotesize
\scriptsize scriptsize \tiny tiny
```

roman, sans serif, SMALL CAPS, typewriter, **bold**, *italics*

Huge huge LARGE Large large normalsize small footnotesize scriptsize tiny

Lists

```
1
2  \begin{itemize}
3    \item Bullet list
4    \item Have no order
5  \end{itemize}
6
7  \begin{enumerate}
8    \item Enumeration
9    \item Make sense, when we want to
       ↪ refer to it by number.
10 \end{enumerate}
11
12 \begin{description}
13   \item[bullet list] Is an unordered
       ↪ list
14   \item[enumeration] Is an ordered
       ↪ list
15 \end{description}
```

- Bullet list
- Have no order

1. Enumeration
2. Make sense, when we want to refer to it by number.

bullet list Is an unordered list

enumeration Is an ordered list

Quotes and footnotes

```
Edsger W.~Dijkstra\footnote{A father
of computer science} said in 2000:

\begin{quote}
   The required techniques of
   effective reasoning are pretty
   formal, \textelp{}
\end{quote}
```

Edsger W. Dijkstra[a] said in 2000:

> *The required techniques of effective reasoning are pretty formal, [...]*

[a] A father of computer science

Language settings

Typesetting depends on the language, e.g., concerning hyphenation. Here comes the `babel` package.

- Many other packages have language options, such as csquotes for using quotes and siunitx for typesetting numbers and units.

```
\usepackage[ngerman]{babel}
% Tell babel to call \sisetup when 'ngerman' locale is set
\addto\extrasngerman{\sisetup{locale = DE}}

\usepackage{siunitx}
\usepackage[autostyle]{csquotes}

\begin{document}
Sind \SI{10.5}{\ampere} eigentlich \enquote{viel} Strom?
\end{document}
```

Math

- We have to switch to math mode for mathematical typesetting.
- For inline math expressions we use `$` as delimiters.
- To put math into a dedicated line we use `\[` and `\]`.

```
Pythogoras says $x^2 + y^2 = z^2$,
Euler says $\int_0^\infty e^{-x}
\, dx = 1$ and De Moivre says
\[ \frac{1}{\sqrt{\pi}}
\int_{-\infty}^\infty e^{-x^2} \,
dx = 1. \]
```

Pythogoras says $x^2 + y^2 = z^2$, Euler says $\int_0^\infty e^{-x}\,dx = 1$ and De Moivre says

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-x^2}\,dx = 1.$$

See [Oet21, chp. 3] for math typesetting. And see sec. 3.10 for symbols, fonts, letters, and so on.

Advanced math

You basically always want to load these packages:

```
\usepackage{amsmath, amsthm, amsfonts, nicefrac, commath}
```

This gives fonts like \mathbb and environments like `align` and `align*`. Some demos:

```
$\nicefrac{\pi}{2} = \arctan 1$ and
\begin{align*}
  (x+y)^2 &= (x+y) \cdot (x+y) \\
          &= x^2 + 2xy + y^2
\end{align*}
and
\begin{align}
  \int_a^b \od{f}{x} \, \dif x =
  f(b) - f(a)
\end{align}
```

$\pi/2 = \arctan 1$ and

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

and

$$\int_a^b \frac{\mathrm{d}f}{\mathrm{d}x}\,\mathrm{d}x = f(b) - f(a) \tag{1}$$

Labels and references

Chapters, sections, figures, tables, equations and so on have numbers, e.g., section 1.1 or figure 2.

We can places labels with `\label` and refer to them using `\ref` and others.

- Label names shall have prefix that tell type of object, e.g., `sec:intro`, `fig:monalisa`, `eq:euler`.

```
\section{Introduction}
\label{sec:intro}

See \eqref{eq:pyth} in
section~\ref{sec:intro} on page
\pageref{eq:pyth}.
\begin{equation}
  a^2 + b^2 = c^2   \label{eq:pyth}
\end{equation}

Use \cref{sec:intro} via package
cleveref.
```

1 Introduction

See (2) in section 1 on page 17.

$$a^2 + b^2 = c^2 \tag{2}$$

Use section 1 via package cleveref.

Figures and graphics

- A figure is a floating body. We tell the figure environment where it shall place it.
- `h` means here, `t` means top, `b` means bottom, `p` means on an own page and `!` means to make LaTeX to try really hard. The order matters.

```
\begin{figure}[!hbtp]
  \centering
  \includegraphics[width=.5\texwidth]{figs/monalisa.jpg}
  \caption{A photography of Mona Lisa.}
  \label{fig:monalisa}
\end{figure}
```

- Within the figure, we include graphics. We typically put them in a subdirectory `figs/`.
- Use package `graphicx` to have `\includegraphics` with advanced options.
- Instead of `includegraphics` there are other means to add figures, e.g., Ti*k*Z.
- There is a `\listoffigures` to typeset the list of figures.

Tables

```
\begin{table}[tb]
  \centering
  \begin{tabular}{lr}
    \toprule
    Item  & Quantity \\
    \midrule
    Apple & 4        \\
    Eggs  & 12       \\
    \bottomrule
  \end{tabular}
  \caption{Shopping list}
  \label{tab:shoppinglist}
\end{table}
```

Item	Quantity
Apple	4
Eggs	12

Table: Shopping list

- Within the floating table, we place the actual tabular.
- We use package `booktabs` for beautiful rules. We avoid vertical lines, see [Fea20].

Line and page breaks

- The command `\\` makes a line break, but no new paragraph.
- The command `\\*` too, but prohibits a page break.
- For vertical space there is `\bigskip`, `\medskip` and `\smallskip`
- The command `\newpage` causes a page break.
- The commands `\clearpage` and `\cleardoublepage` causes a page break (resp. for two-sided documents) but also places all remaining floating objects.

Compilers

There are different LaTeX distributions.

- But *TeX Live* became the standard distribution in the last decade.

There are different compilers:

- Decades ago we used `latex` to produce DVI files, and converted them to postscript or PDF.
- Then we just used `pdflatex`. Today we use `lualatex`.

LuaTeX:

- Stable since 2016. Part of Tex Live since 2008.
- Provides `lualatex` compiler.
- Unicode as input encoding. We can write `ä` instead of `\"a`.
- Also supports OpenType and TrueType fonts. (Before that we needed XeTeX.)

Editors and IDEs

There are many LaTeX IDEs, including:

- Texmaker
- TeXstudio
- Visual Studio Code

Some people use Overleaf for online collaboration.

For vi users:

- Definitely look at the vimtex plugin
- You may like latex-unicoder
- There is tree-sitter alpha support

For Linux users and beyond:

- Definitely look at `latexmk` to build latex documents.

Bibliography basics

- Bib files contain references as entries.
- There are many different entry types, e.g., `@book`, `@article`, `@inproceedings`, `@online`, `@phdthesis`, `@techreport`, `@unpublished`, `@patent`, `@misc`
- Different entry types support different fields, e.g., `@article` knows `volume`.
- People build entire databases of bib files over years. Scientists sometimes publish bib files of their publications.
- Journals and article databases support exporting bib entries.
- In the tex file we refer to the bib files and cite the reference using the `\cite` command.

```
@book{
  Sedgewick2013
  author = {Sedgewick, Robert},
  title = {Algorithms in C++},
  publisher = {Addison-Wesley},
  isbn = {978-0321606334},
  year = 2013
}
```

```
A great algorithms book
is \cite{Sedgewick2013}.
```

Bibliography usage

Today we use biber as engine for bib files.[1]

- Supports unicode, multiple bibliographies, filtering, et cetera.
- It is used with `biblatex` for LaTeX. See [KWL22] for documentation.

```
\usepackage[
  backend=biber,
  % Many citation styles: numeric,
      ↪ authoryear, ieee, apa,
      ↪ nature, ...
  style=alphabetic,
  ]{biblatex}
\addbibresource{literature.bib}
```

```
\section{Introduction}

I like \cite{Sedgewick2013} by
\citeauthor{Sedgewick2013} as
algorithm book.

% Print the bibliography
\printbibliography
```

- Use `\nocite` to add reference without citing it.

[1] Historically, it was bibtex.

Bibliography management

There is software, like JabRef, to organize bib files. I edit them in vi and organize them as plain bib files.

JabRef can also attach the PDF of a reference. I organize my digital library as a directory structure.

Creating figures

High quality figures are key to your papers, presentations, theses, and so on.

I *never*

- create figures in Word, Powerpoint, Visio or such or
- put pictures from websites, screenshots or scans in my work.

Instead

- put a lot of thought in what your figure shall tell, display, emphasize, illustrate and
- match the font sizes, (mathematical) notations, and so on with the surrounding text

I personally have been using three tools for figures:

- In 85 % I use Ipe
- In 14 % I use Ti*k*Z
- In 1 % I use Inkscape or Krita

Ipe editor

Ipe is a vector graphics editor

- Text is done in LaTeX
- File format is PDF, hence friendly for `\includegraphics`
- Layers in Ipe translate to PDF pages, which can be used for animations in LaTeXpresentations through beamer overlays
- CAD-like features like snapping on line intersections, endpoints and such
- Ipelets can be scripted in lua. Geometry library CGAL provides extensive ipelet.
- With style files we can manage styles that match with the document

Further resources:

- Ipe website: `https://ipe.otfried.org/`
- Ipe wiki: `https://github.com/otfried/ipe-wiki/wiki`
- My personal Ipe webpage: `https://www.sthu.org/misc/ipe.html`

TikZ

TikZ (or rather PGF/TikZ) is a programming language for graphics:

- Comes with an extensive manual [Tan21] and a shorter version [Cré11].
- Produces high-quality figures, but is laborious.
- Have a look at examples at `https://en.wikipedia.org/wiki/PGF/TikZ` and `https://texample.net/tikz/examples/`

```
\begin{figure}
  \centering
  \begin{tikzpicture}[]
    \draw[blue] (-2,-1) -- (-2,1);
    \foreach \i in {1, ..., 13}
      \foreach \j in {\i, ..., 13}
        \draw (\i*360/13:1.5)
          -- (\j*360/13:1.5);
  \end{tikzpicture}
  \caption{Blue line and $K_{13}$.}
\end{figure}
```

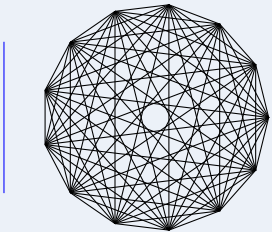
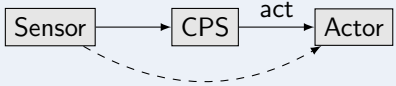


Figure: Blue line and K_{13}.

A key concept in TikZ are nodes.

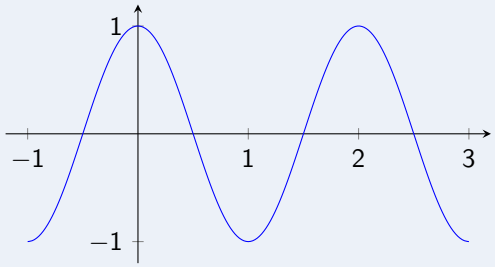
```
\begin{tikzpicture}[
    box/.style={draw,fill=black!10}]

  \node[box] (s) {Sensor};
  \node[box, right=of s] (c) {CPS};
  \node[box, right=of c] (a) {Actor};
  \draw[->] (s) -- (c);
  \draw[->] (c)
    -- node[above] {act} (a);
  \draw[->,dashed] (s)
    to[bend right] (a);
\end{tikzpicture}
```



TikZ and plots

```
\begin{tikzpicture}
    \begin{axis}[%
      axis lines=middle,
      enlargelimits={abs=0.2},
      height=5cm, width=8cm,
    ]
    \addplot[domain=-1:3, samples=100,
      smooth, blue]
      {cos(deg(pi*x))};
  \end{axis}
\end{tikzpicture}
```



Alternatively, I use matplotlib in jupyter notebooks for high-quality plots.

- However, take care to export vector graphics (e.g., PDF, SVG), not PNGs or – for god's sake – JPGs.
- There are ways to have TikZ output from matplotlib.

Presentations

Presentations in LaTeX today are done with package `beamer`, see [WMT].

- *All* my slides are in beamer.

```
\documentclass[aspectratio=169]{beamer}
\title{\LaTeX\ and friends}
\author{Stefan Huber}
\institute{FH Salzburg}
\date{Summer 2022}

\begin{document}

\frame{\titlepage}

\begin{frame}{Introduction}
  Hello world.
\end{frame}

\end{document}
```

siunitx

For natural and engineering sciences, we often handle physical quantities and units. Typesetting is subtle here:

```
$2 cm/kg\,s$ is the formula
$2\frac{c \cdot m}{k \cdot g}$, but
$\SI{2}{\centi\meter\per\kilo\gram}$
is a physical quantity.
```

$2cm/kg\,s$ is the formula $2\frac{c \cdot m}{k \cdot g}$, but $2\,\mathrm{cm\,kg^{-1}}$ is a physical quantity.

The package `siunitx` is for typesetting numbers, angles, percents, units, complex numbers and quantities, see [Wri22].

```
\SI{40}{\percent}, \ang{30},
\ang{45;30;4}, \num{2e6}, \num{e80}
```

$40\,\%$, 30°, $45^\circ 30' 4''$, 2×10^6, 10^{80}

siunitx

It adds a column type S to align numbers:

```
\begin{tabular}{S}
  \toprule
  {Numbers}  \\
  \midrule
  0.012      \\
  12.01      \\
  .34        \\
  345        \\
  1.256e-6   \\
  \bottomrule
\end{tabular}
```

Numbers
0.012
12.01
0.34
345
1.256×10^{-6}

Personal remark

There is a pandemic on wrong typesetting of physical units, e.g., [kg] is wrong. See https://www.sthu.org/blog/18-notation-phys-einheiten/ for details.

Glossaries and acronyms

Use of glossaries and acronyms and summary table generation is supported by the package `glossaries`, see [Tal21].

- Acronyms is just for definition of the long version, e.g., PC stands for Personal Computer.
- Glossary is for definition and description of terms, e.g., a PC is a computer for personal use unlike, say, a server or mainframe computer.

```
\usepackage[toc,nonumberlist,
    ↪ nopostdot,style=super]{
    ↪ glossaries}
%\usepackage{glossaries-extra}
\loadglsentries{glossary-defs}
\makeglossaries
```

The `glossary-defs.tex` file:

```
\newacronym{ai}{AI}{Artificial
    ↪ Intelligence}
\newacronym{ann}{ANN}{Artificial
    ↪ Neural Network}
```

Glossaries

- On the first use of an acronym, the long version is printed.
- This is handy for a paper, where we would not print a glossary table. So here, we could disable this first-use feature.

```
% Only prints the glossary entries
% that are actually used
\printglossaries

\section*{Introduction}
We see \glspl{ann} as a method of
\gls{ai}. But \gls{ai} has many
more methods.
```

Glossary

AI	Artificial Intelligence
ANN	Artificial Neural Network

Introduction

We see Artificial Neural Networks (ANNs) as a method of Artificial Intelligence (AI). But AI has many more methods.

Further packages

geometry	To adjust page geometries, like margins and more.
listings	For code listings
currency	To typeset currencies
rotating	For rotated pages
gitinfo2	To place git infos within document, like document version from git tag
fancyhdr	For advanced, customized page headers
xspace	To use the right space at certain situations
hyperref	To have hyperlinks at references
xcolor	For colored text
titlesec	For fancy chapter headings
longtable	For multi-page tables
enumitem	For more customized enumerations and bullet lists
marginnote	For margin notes
lineno	For line numbers
makeidx	To generate indices
subcaption	For subfigures in figures
algpseudocode	For algorithms in pseudocode

Concluding remarks

LaTeX is the standard for high-quality text.

- It is the standard for all MINT disciplines and beyond.
- I use it also for important letters, like job applications.

There is a git repo with a LaTeX template for FHS theses:

- `https://its-git.fh-salzburg.ac.at/FB-Mechatronik/fhs-its-thesis-latextemplate`
- Such a template reliefs you from the burden to setup all basic infrastructure, like glossaries, situnix, and so on. So you can start, more or less, right away.
- You find some links to learn LaTeX in the README.
- Although it is not official, I kind of maintain it.

Some more resources

- Martin Held, University of Salzburg:
  `https://www.cosy.sbg.ac.at/~held/teaching/wiss_arbeiten/`
- Learn LaTeX in 30 minutes:
  `https://de.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`
- A beginner's guide to LaTeX (7 pages):
  `https://www.cs.princeton.edu/courses/archive/spring18/cos423/latex/latex-guide-cos423.pdf`
- *The Not So Short Introduction to LaTeX 2ε* (153 pages): [Oet21]
- LaTeX Beginner's Guide (336 pages):
  `http://static.latexstudio.net/wp-content/uploads/2015/03/LaTeX_Beginners_Guide.pdf`

Bibliography I

[Cré11] Jacques Crémer. *A very minimal introduction to TikZ*. Mar. 2011. URL: http://cremeronline.com/LaTeX/minimaltikz.pdf.

[Fea20] Simon Fear. *Publication quality tables in LaTeX*. Jan. 2020. URL: http://mirrors.ctan.org/macros/latex/contrib/booktabs/booktabs.pdf.

[KWL22] Philip Kime, Moritz Wemheuer, and Philipp Lehman. *The biblatex package. Programmable Bibliographies and Citations*. Version 3.17. Feb. 2022. URL: https://mirror.easyname.at/ctan/macros/latex/contrib/biblatex/doc/biblatex.pdf.

[Oet21] Tobias Oetiker. *The Not So Short Introduction to LaTeX 2ε*. Mar. 2021. URL: https://tobi.oetiker.ch/lshort/lshort.pdf.

[Tal21] Nicola L.C. Talbot. *The glossaries package v4.49: a guide for beginners*. Nov. 2021. URL: http://tug.ctan.org/macros/latex/contrib/glossaries/glossariesbegin.pdf.

[Tan21] Till Tantau. *TikZ and PGF*. Version 3.1.9a. May 2021. URL: https://mirror.kumi.systems/ctan/graphics/pgf/base/doc/pgfmanual.pdf.

[WMT] Joseph Write, Vedran Miletić, and Till Tantau. *The beamer class*. Version 3.66. URL: http://tug.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf.

[Wri22] Joseph Wright. *siunitx – A comprehensive (SI) units package*. Feb. 2022. URL: http://ctan.math.utah.edu/ctan/tex-archive/macros/latex/contrib/siunitx/siunitx.pdf.