



L^AT_EX ON OVERLEAF

A GUIDE

by GABRIEL CABANA

V1.0

2021-08-26

CHANGELOG

Version	Date	Description
0.1	2021-05-05	First draft: <ul style="list-style-type: none">– Introduction;– Text formatting macros;– Referencing macros;– List examples;– Figure examples;– Special characters;– Custom macros.
0.2	2021-05-13	<ul style="list-style-type: none">– Split document in multiple \TeX files and sections;– Added math macros and equation examples;– Added table examples.
0.3	2021-05-14	Added templates: <ul style="list-style-type: none">– Equations;– Figures;– Lists;– Tables.
0.4	2021-05-15	Added section and template for acronyms.
0.5	2021-05-27	<ul style="list-style-type: none">– Split cover and table of contents (ToC) in different pages;– Added background image to cover page, ToC, and lists;– Added list of figures (LoF);– Added list of tables (LoT);– Empty sections for architecture and preamble.

Version	Date	Description
0.6	2021-05-31	<ul style="list-style-type: none"> – Regrouped sections into new “\LaTeX basics” section; – \LaTeX on Overleaf section: <ul style="list-style-type: none"> • Engine; • Format; • Distribution; • Editor. – Architecture section: <ul style="list-style-type: none"> • Root files; • Custom packages folder; • Data folder; • Custom definitions folder; • Sections folder; • Tools folder. – Templates section: <ul style="list-style-type: none"> • Equations; • Figures; • Lists; • Tables; • TikZ drawings.
0.7	2021-06-04	<ul style="list-style-type: none"> – Added bibliography; – \LaTeX basics section: <ul style="list-style-type: none"> • Special characters; • Text formatting and sizing; • Sections; • Links and references; • Lists; • Mathematical equations; • Figures; • Tables; • Custom macros.

Version	Date	Description
0.8	2021-06-22	<ul style="list-style-type: none"> – Added custom column widths in “Custom macros” section; – Added special letters in “Special characters” section; – Added spacing macros in “Text” section; – Added code listings in “L^AT_EX basics” section; – Modified tables for better column widths; – Added TikZ templates; – Modified global code listing style; – Added custom acronym / glossary list style; – Added L^AT_EX language for code listings; – Added list of listings (LoL).
0.9	2021-08-22	<ul style="list-style-type: none"> – Added file types in “L^AT_EX on Overleaf” section; – Added <code>loadgeometry</code> command for landscape; – Added more macros for glossary and acronym entries; – Added templates for sources; – Added templates for glossary and acronym entries; – Preamble section: <ul style="list-style-type: none"> • Document class; • Footnotes; • Language; • Bibliography.

Version	Date	Description
0.10	2021-08-23	<p>Preamble section:</p> <ul style="list-style-type: none"> – Date and time; – Fonts; – Geometry; – Typesetting; – ToC and lists; – Figures; – Colors; – TikZ; – Text; – Conditionals; – Header and footer; – Tables; – Mathematics and arithmetic; – Hyperlinks; – Counters and numbering; – Sections; – Lists; – Glossary; – Custom packages; – Miscellaneous.
1.0	2021-08-26	<ul style="list-style-type: none"> – Revision of the ToC style; – Revision of sources; – Added <code>hologo</code> package for \TeX logos; – Empty section for “\TeXnician basics”; – Added glossary to set the terminology; – Revision of the page breaks and spacing between sections.

CONTENTS

Acronyms	xiv
Foreword	xv
<hr/>	
1 L^AT_EX on Overleaf	1
1.1 Engine	2
1.2 Format	3
1.3 Distribution	3
1.4 Editor	4
1.5 File types	4
2 L^AT_EX basics	7
2.1 Special characters	7
2.2 Dimensions	9
2.3 Text	10
2.4 Sections	18
2.5 Links and references	19
2.6 List	23
2.7 Math	24
2.8 Figure	27
2.9 Table	28
2.10 Typesetting	31
2.11 Graphics	32
2.12 Custom macros	32
2.13 Debugging tools	35
3 T_EXnician basics	37
<hr/>	
4 Architecture	38
4.1 Root files	39
4.2 cmds/	40
4.3 data/	41
4.4 defs/	41

4.5	src/	42
4.6	tools/	44
5	Preamble	45
5.1	Document class	45
5.2	Footnotes	46
5.3	Language	47
5.4	Bibliography	48
5.5	Date and time	51
5.6	Fonts	51
5.7	Geometry	52
5.8	Typesetting	55
5.9	Table of contents and lists	57
5.10	Figures	58
5.11	Colors	59
5.12	TikZ	60
5.13	Text	62
5.14	Conditionals	63
5.15	Header and footer	63
5.16	Tables	64
5.17	Mathematics and arithmetic	66
5.18	Hyperlinks	68
5.19	Counters and numbering	70
5.20	Sections	70
5.21	Lists	71
5.22	Glossary	71
5.23	Custom packages	73
5.24	Miscellaneous	73
<hr/>		
6	Templates	76
6.1	Equations	76
6.2	Lists	78
6.3	Figures	79
6.4	Tables	80
6.5	TikZ	88

6.6	BIB _T E _X	93
6.7	Glossary	101
6.8	Acronyms	101

Sources	102
Glossary	105

LIST OF FIGURES

4.1	Global architecture	38
4.2	Root files	39
4.3	Architecture of packages	40
4.4	Comparison of section file and data file architectures	41
4.5	Architecture of section files	42
4.6	Architecture of template files	43
4.7	Architecture of tool files	44
5.1	Document layout information	54
<hr/>		
6.1	Template: single figure	79
6.2	Template: figure with subfigures	80
6.3	Template: TikZ #1	88
6.4	Template: TikZ #2	88
6.5	Template: TikZ #3	88
6.6	Template: TikZ #4	89
6.7	Template: TikZ #5	89
6.8	Template: TikZ node referencing	91

LIST OF TABLES

1.1	\TeX architecture	1
1.2	\TeX input file types	5
1.3	\TeX output file types	6
2.1	\LaTeX reserved characters	7
2.2	\LaTeX diacritics	7
2.3	\LaTeX special letters	8
2.4	\LaTeX typographic dimensions	10
2.5	\LaTeX text formatting (family)	11
2.6	\LaTeX text formatting (series)	11
2.7	\LaTeX text formatting (shape)	11
2.8	\LaTeX text formatting (other)	11
2.9	\LaTeX text size commands	12
2.10	\LaTeX text size (article, report, book, letter)	13
2.11	\LaTeX text size (memoir)	13
2.12	\LaTeX text size (beamer)	14
2.13	\LaTeX horizontal spacing commands	15
2.14	\LaTeX vertical spacing commands	17
2.15	\LaTeX section levels	18
2.16	\LaTeX labelling examples	20
2.17	\LaTeX mathematical environments	24
2.18	\LaTeX mathematical commands	25
2.19	\LaTeX float specifiers	27
2.20	\LaTeX subfigure alignment specifiers	27
2.21	\LaTeX table environments	29
2.22	\LaTeX table columns	30
2.23	Custom table columns	33
<hr/>		
6.1	Template: simple table	80
6.2	Template: table with merged rows and columns	82
6.3	Template: landscape table	84
6.4	Template: table with multi-line cells and lists	86
6.5	Template: TikZ colored table	90

6.6	Template: TikZ table with alternating row colors	92
6.7	Template: TikZ fancy table	92

LIST OF LISTINGS

2.1	\LaTeX listing environment	32
2.2	Column length assignment (I)	34
2.3	Column length assignment (II)	34
2.4	Multi-column with custom column widths	35
2.5	Display length in text	35
<hr/>		
5.1	Document class settings	45
5.2	Footnotes settings	46
5.3	Language settings	47
5.4	Language options	47
5.5	Bibliography settings	48
5.6	Bibliography options	49
5.7	Date and time settings	51
5.8	Font settings	51
5.9	Geometry settings	52
5.10	Listing settings	55
5.11	Listing style	55
5.12	Table of contents and lists settings	57
5.13	Figure settings	58
5.14	Color settings	60
5.15	TikZ settings	60
5.16	Text settings	62
5.17	Conditionals settings	63
5.18	Header and footer settings	63
5.19	Header and footer style	63
5.20	Table settings	64
5.21	Mathematics and arithmetic settings	66
5.22	Hyperlink settings	68
5.23	Counters and numbering settings	70
5.24	Sections settings	70
5.25	Lists settings	71
5.26	Glossary settings	71

5.27	Glossary style	72
5.28	Custom packages	73
5.29	Miscellaneous settings	74
<hr/>		
6.1	Template: single equation	76
6.2	Template: set of equation	76
6.3	Template: mathematical development	77
6.4	Template: unordered list	78
6.5	Template: ordered list	79
6.6	Template: single figure	79
6.7	Template: single figure, top caption	79
6.8	Template: figure with subfigures	80
6.9	Template: simple table	80
6.10	Template: table with merged rows and columns	82
6.11	Template: landscape table	85
6.12	Template: table with multi-line cells and lists	86
6.13	Template: article source	93
6.14	Template: in proceedings source	94
6.15	Template: book source	95
6.16	Template: in book source	96
6.17	Template: thesis source	97
6.18	Template: report source	98
6.19	Template: website source	99
6.20	Template: web document source	99
6.21	Template: legislation source	100
6.22	Template: norm source	100
6.23	Template: glossary entry	101
6.24	Template: acronym entry (native language)	101
6.25	Template: acronym entry (foreign language)	101
6.26	Template: acronym entry (proper noun)	101

ACRONYMS

AAT	Apple Advanced Typography	52
ASCII	American Standard Code for Information Interchange	7
CTAN	Comprehensive T _E X Archive Network	3, 22, 23, 32, 45, 57, 58, 62, 69, 102, 103
DPI	Dots Per Inch	10
DVI	Device Independent File	2, 6
FOSS	Free and Open-Source Software	3
IDE	Integrated Development Environment	1
ISO	International Organization for Standardization	49, 51, 103
LoF	List of Figures	ii, 6, 28, 58
LoL	List of Listings	iv, 58
LoT	List of Tables	ii, 6, 58
PDF	Portable Document Format	1–3, 6, 10, 39, 41, 69, 74, 75, 102, 103
PGF	Progressive Graphics File	28, 32, 103, 104
RC	Run Commands	39, 103
ToC	Table of Contents	ii, v, 6, 22, 57, 58, 69, 72, 104
TUG	T _E X Users Group	102
URL	Uniform Resource Locator	20, 21, 49, 69, 106
WYSIWYG	What You See Is What You Get	xv, 10, 28
WYSIWYG	What You See Is What You Mean	xv, 10
XML	Extended Markup Language	6

FOREWORD

Contrary to what we're used to with Microsoft Word, Apache OpenOffice Writer, Google Docs, or Apple Pages, \LaTeX doesn't display the text as it will appear in the final document when you're writing it. They use the what you see is what you get (WYSIWYG) approach, while \LaTeX use what you see is what you mean (WYSIWYG) (appearance vs structure-based approach) Is it plain text, that uses code, or macros, to change its appearance and behavior. Tables and figures must be added with macros too: they even have their own environment so that their special behavior is isolated from the rest of the document (`\begin{<env>}\dots\end{<env>}`). It makes it a pain to setup, but once it's done, it is a powerful tool that can help automate your writing by:

- automatically setting up the bibliography;
- writing acronyms and technical words in different ways depending on how much you want to clue in the reader, and put them in a list of acronyms and/or a glossary;
- add conditionals, loops, and hooks;
- draw graphics;
- use arithmetic functions to work out the result of a formula;
- and much more.

It also allows us to sync the files with a version control software to get a much more appealing history of modifications compared to Word's markup.

For you, it will be a *little* harder than writing with Word since you have to use code to modify your text. Fear not, all the major macros needed for our reports will be listed here. You can also skim through the files to see how they can be used or look at the templates (see § 6).

Disclosure: This guide is written *to the best of my knowledge*. If some information becomes deprecated, some errors are found, or a better way to solve a certain issue can be added to the project, let me know.

1 L^AT_EX ON OVERLEAF

This project was implemented in Overleaf. It is not necessary to understand the section, although it provides examples of what is discussed. For proper illustration of my words, start a project with your own Overleaf account and follow along, or ask for viewing privileges on this project if you know me personally.

If you go over the Overleaf menu of this project, most settings will be straightforward to you, especially those that have already programmed and are familiar working with an integrated development environment (IDE). You will see settings such as “spell check”, “code check”, “auto-completion”, etc. They are pretty self-explanatory. All fine and dandy, until you get to “compiler” and “T_EX Live version”:

- What’s T_EX Live?
- What T_EX Live version should I use? Always the most up-to-date?
- What compiler should I choose for my project?
- What distinguishes the different compilers?

Fear not. Most of the different options can be found in table 1.1, and then some. Each level will be described in more details in the following subsections. More information is also available on Overleaf [1]. For simplicity, “L^AT_EX” may be used pretty loosely in the rest of the document to refer to the typesetting system in part or in its entirety.

Table 1.1 T_EX architecture.

Level	Examples	Description
Engine	<ul style="list-style-type: none">– T_EX– pdfT_EX– X_YT_EX– LuaT_EX– . . .	These are the executable binaries which implement different T _E X variants, known as the “compiler” option in Overleaf.

Level	Examples	Description
Format	<ul style="list-style-type: none"> – \TeX – \LaTeX – \OpTeX – ... 	These are the \TeX -based languages in which one actually writes documents. When someone says “ \TeX is giving me a mysterious error”, they usually mean a format. If no special format is defined, one can write in plain \TeX .
Distribution	<ul style="list-style-type: none"> – \TeX Live – \MiKTeX – ... 	These are the large, coherent collections of \TeX -related software to be downloaded and installed. When someone says “I need to install \TeX on my machine”, they’re usually looking for a distribution.
Editor	<ul style="list-style-type: none"> – Overleaf – \LyX – Vim – ... 	These editors are what you use to create or modify a document file.

1.1 ENGINE

Overleaf offers four different engines to power our projects. Each one builds on its predecessor to add functionality. More information is available on Overleaf [2].

- \TeX : the basis of the entire typesetting system. Its first release dates back to 1978. It publishes the compiled document using the device independent file (DVI) format, which could later be processed to convert the document to PostScript. Overleaf refers to \TeX as “ \LaTeX ”, though the latter is less of an engine flavor and more of a higher-level programming language—or format—that builds on top of plain \TeX and includes nice amenities. It’s all semantics though, and I don’t know the inner workings. I’m just trying to provide a nice way to think about the software layers;
- \pdfTeX : implements direct PDF output, along with a variety of extensions;
- \XeTeX : does everything \pdfTeX can do, while also supporting Unicode, OpenType fonts, TrueType fonts, and system fonts natively;
- \LuaTeX : does everything \XeTeX can do, while also adding a programming layer written in Lua that gives access to many of the internal settings of the \TeX engine.

There are other engines, but the above are by far the most common ones. For this template, every-

thing was verified and compiled in XeTeX . By definition, LuaTeX should work as well, but might be more bloated than necessary since we don't play with internal parameters—yet. TeX or pdfTeX will not work.

For more details about TeX , please visit Donald E. Knuth's books about the intent and inner workings of his system [3–5].

1.2 FORMAT

The format can be understood as a higher-level language to write TeX commands. There is plain TeX , which offers no level of abstraction: you're writing with the same macros that were defined in the 70s and 80s. There is also $\text{L\text{a}TeX}$, which is the most common language and the one used by Overleaf. It was created in the 90s to provide flexibility, extension capabilities, and abstraction from plain TeX . For instance, when you're starting an environment, $\text{L\text{a}TeX}$ translates your simple command into TeX macros for the engine to interpret. There are other formats available, such as OpTeX , though I am much less familiar with those and would prefer not to misguide you in this document. You can find more information about OpTeX in its Comprehensive TeX Archive Network (CTAN) documentation [6].

1.3 DISTRIBUTION

Distributions can regroup large collections of macros, packages, and classes into a single installer. The most common ones are MiKTeX and TeX Live . More and more people are leaning towards TeX Live over time, as it is maintained by a community of users instead of a single person or a small group: it can stand the test of time more easily. MiKTeX can offer some features that are not available in the other distribution or are better implemented, so the distribution you select for a local install is entirely up to you and your depth of knowledge. Overleaf uses TeX Live on their servers, so we don't have a choice when writing documents online.

TeX Live is intended to be a straightforward way to get up and running with the TeX document production system. It provides a comprehensive TeX system with binaries for most flavors of Unix, including GNU/Linux, macOS, and also Windows. It includes all the major TeX -related programs, packages, fonts that are free and open-source software (FOSS), and support for many languages [7]. It has been in development since 1996, and a new version comes out every year, integrating new features and bug fixes. TeX in general moves slowly, so you can stay on whatever

version of T_EX Live your document was created with—unless you need the updated utilities of a newer version. The present project was tested with T_EX Live 2020.

1.4 EDITOR

If you're writing a document locally, you can use any editor of your choice: L^AT_EX, Emacs, Vim, Notepad++... Being bound to the realm of the World Wide Web, and in our cozy, no-command-line world, we will stick with Overleaf.

If you hear the name “ShareL^AT_EX” being thrown around, they merged with Overleaf a couple years ago, and their documentation and accounts carried over. There was also “WriteL^AT_EX”, although I have no information about that one—I suspect it was Overleaf's previous name.

The basics of an Overleaf project include:

- theming, to customize source code colors and syntax highlighting the way you like it;
- choice of engine and, although limited, distribution version;
- synchronization with Dropbox, Git or GitHub;
- code auto-completion;
- code / text review;
- multi-user editing and sharing;
- gallery publishing;
- history logging;
- in-project messaging;
- rich text mode.

1.5 FILE TYPES

Every step in the document compilation chain can use or generate files: the engine, the format, the packages, etc. Most types of input and output files can be found in tables 1.2 and 1.3.

Most Overleaf users will only need to worry about the highlighted files. As you get on your way to become a T_EXnician, or if you start running your projects locally, you may start looking into more and more file types (see § 3).

Table 1.2 \TeX input file types.

File	Extension	Description
Source	<code>.tex</code>	General purpose file. It may include text, symbols, mathematical expressions, graphics, and more.
Class	<code>.cls</code>	File that stores a document class with predefined typeset configuration. It can be used to create custom articles, books, posters, etc.
Class options	<code>.clo</code>	File that stores a document class' options. Usually, they are defined in a <code>.cls</code> file, but they are occasionally defined separately.
Style	<code>.sty</code>	File that contains macros of specific styling attributes.
Definition	<code>.def</code>	File that contains lists of commands and definitions that would clog a <code>.sty</code> file.
Sources	<code>.bib</code>	Database that stores sources for the bibliography.
Bib \TeX style	<code>.bst</code>	File that contains styling attributes for the bibliography.
Bibliography style	<code>.bbx</code>	File that contains styling attributes for the bibliography generated by the <code>biblatex</code> package.
Cite style	<code>.cbx</code>	File that contains styling attributes for citations generated by the <code>biblatex</code> package.
Documented source	<code>.dtx</code>	File that contains both content and documentation. Useful to generate a \LaTeX package with its associated documentation.
Installation	<code>.ins</code>	File that stores instructions to extract template files out of a <code>.dtx</code> file.
Configuration	<code>latexmkrc</code>	File that stores command line arguments for automatic document compilation.

Table 1.3 \TeX output file types.

File	Extension	Description
DVI	.dvi	The original output file for documents when \TeX was created. They consist of binary data describing the visual layout of a document in a manner not reliant on any specific image format, display hardware or printer.
PDF	.pdf	The default document output file for pdf \TeX , X \TeX , Lua \TeX . For all practical purposes, you should always compile with this type of output.
Auxiliary	.aux	File that saves information for the implementation of references, footnotes, and bibliographies, among other things.
Log	.log	File that stores all messages of the compilation, like errors and warnings.
Font definition	.fd	File that contains the font information used to generate the document output.
Bib \TeX	.bbl	Bib \TeX output for insertion into the document output.
Bib \TeX log	.blg	File that contains compilation messages specific to Bib \TeX .
Bibliography control	.bcf	Control file for the bibl \TeX package.
Bibliography XML	.run.xml	Extended markup language (XML) file for the bibliography engine biber.
Table of contents	.toc	Auxiliary file that contains the ToC.
List of figures	.lof	Auxiliary file that contains the LoF.
List of tables	.lot	Auxiliary file that contains the LoT.
Glossary	.glo	Glossary file used to generate the document output.
Glossary log	.glg	File that contains compilation messages specific to glossaries.

2 L^AT_EX BASICS

2.1 SPECIAL CHARACTERS

Like every programming language, L^AT_EX reserves some characters for very specific purposes. They can be included in text with either an escape sequence or a special command. Although this should not be an issue with X_YL^AT_EX or LuaT_EX, some T_EX engines also have a hard time dealing with non-ASCII input, thus we also need to escape some diacritics.

Table 2.1 L^AT_EX reserved characters.

Character	Escape Sequence	Description
&	\&	Used for alignment in certain environments.
%	\%	Used mostly for comments.
\$	\\$	Used for inline math.
#	\#	Used for parameters in macro declarations.
{	\{	Used for grouping.
}	\}	Used for grouping.
_	_	Used for subscripts in math mode.
^	\textasciicircum	Used for superscripts in math mode.
~	\textasciitilde	Used for non-breakable space.
\	\textbackslash	Used to declare macros.

Table 2.2 L^AT_EX diacritics.

Character	Escape Sequence	Description
ò	\`{o}	Grave accent.
ó	\'{o}	Acute accent.
ô	\^{o}	Circumflex accent.

Character	Escape Sequence	Description
ö	\o{}	Umlaut, trema or dieresis.
ő	\H{o}	Hungarian umlaut (double acute accent).
õ	\~{o}	Tilde.
ç	\c{c}	Cedilla.
ą	\k{a}	Ogonek.
ō	\={o}	Macron.
ȕ	\b{o}	Underline.
ó	\. {o}	Dot over the letter.
ȝ	\d{o}	Dot under the letter.
å	\r{a}	Ring over the letter.
ö	\u{o}	Breve over the letter.
č	\v{c}	Caron or háček over the letter.
ôa	\t{oa}	Tie over two letters.

Table 2.3 L^AT_EX special letters.

Character	Escape Sequence	Description
œ / Œ	\oe{} / \OE{}	French ligature “oe”.
æ / Æ	\ae{} / \AE{}	Latin or scandinavian “ae”.
å / Å	\aa{} / \AA{}	Scandinavian “a” with circle.
ł / Ł	\l{} / \L{}	Barred “l”.
ø / Ø	\o{} / \O{}	Slashed “o”.
ı	\i{}	Dotless “i”.
ȝ	\j{}	Dotless “j”.
ß	\ss{}	German sharp “s”.

Character	Escape Sequence	Description
§	<code>\S{ }</code>	Section mark.
¶	<code>\P{ }</code>	Paragraph mark.
†	<code>\dag{ }</code>	Dagger.
‡	<code>\ddag{ }</code>	Double dagger.

2.2 DIMENSIONS

Dimensions are useful to specify or create new length commands. \TeX engines give you a lot of options to accommodate your needs. Conversions to most of the available dimensions can be found in table 2.4.

It's important to know what the different options are for. Some are relative, like `ex` or `em`: they're better suited for text manipulation. Most are set in stone, like `in` or `cm`: they might fit invariant elements better. In any case though, you should know that \TeX uses fixed point arithmetic, and thus precision issues start to appear when you need five significant digits or more. The smaller the unit used, the lesser of a problem it becomes.

The available units are:

- `sp`: scaled point, the smallest \TeX unit (1 `sp` = 1/65536 `pt`);
- `pt`: point, smallest unit of measurement in typography (1 `pt` = 1/72.27 `in`);
- `bp`: big point, also known as the PostScript point (1 `bp` = 1/72 `in`). Most used as a normal point in other software suites;
- `dd`: Didot point (1157 `dd` = 1238 `pt`);
- `mm`: millimeter;
- `ex`: nominal height or the distance between the baseline and the mean line of lower-case letters;
- `em`: nominal width corresponding to the currently specified point size;
- `pc`: pica (1 `pc` = 12 `pt`);
- `cc`: cicero (1 `cc` = 12 `dd`);
- `cm`: centimeter;
- `in`: inch (1 `in` = 72 `bp`);
- `mu`: math unit (1 `em` = 18 `mu`, where `em` is taken from the math symbols family). Various lengths are derived from it (see § 2.3.3);

- px: available when using the pdf \TeX or Lua \TeX engines (see § 1.1), it corresponds to the inverse of the dots per inch (DPI) resolution.

Table 2.4 \LaTeX typographic dimensions.

	pt	bp	dd	mm	ex	em	pc	cc	cm	in
pt	1.00000	1.00375	1.07001	2.84528	3.60001	8.00000	12.00000	12.84010	28.45276	72.27000
bp	0.99626	1.00000	1.06601	2.83465	3.58656	7.97011	11.95517	12.79213	28.34646	72.00000
dd	0.93457	0.93808	1.00000	2.65911	3.36446	7.47658	11.21486	12.00000	26.59115	67.54151
mm	0.35146	0.35278	0.37607	1.00000	1.26526	2.81168	4.21752	4.51278	10.00000	25.40000
ex	0.27778	0.27882	0.29722	0.79035	1.00000	2.22222	3.33333	3.56669	7.90353	20.07497
em	0.12500	0.12547	0.13375	0.35566	0.45000	1.00000	1.50000	1.60501	3.55659	9.03375
pc	0.08333	0.08365	0.08917	0.23711	0.30000	0.66667	1.00000	1.07001	2.37106	6.02250
cc	0.07788	0.07817	0.08333	0.22159	0.28037	0.62305	0.93457	1.00000	2.21593	5.62846
cm	0.03515	0.03528	0.03761	0.10000	0.12653	0.28117	0.42175	0.45128	1.00000	2.54000
in	0.01384	0.01389	0.01481	0.03937	0.04981	0.11070	0.16604	0.17767	0.39370	1.00000

2.3 TEXT

Unlike most word processors that work on the WYSIWYG principle, \LaTeX with its WYSIWYG approach needs you to specify the text characteristics with commands. You can specify the format, the size, and the spacing.

2.3.1 FORMATTING

Text formatting can be split into three categories:

- family: defines the font type;
- series: defines the line thickness;
- shape: defines the form of the letters.

Note that **YOU CAN COMBINE THEM ALL AS YOU WISH**, though the result might not be aesthetically pleasing. Also note that not all fonts support the various options, and some of them are not orthogonal within a font: some options will conflict and the resulting text won't look like you expected. Tables 2.5 to 2.8 document the most common text commands.

Table 2.5 L^AT_EX text formatting (family).

Command	Example	Description
<code>\textrm{ }</code>	Text	Roman text.
<code>\textsf{ }</code>	Text	Sans serif text.
<code>\texttt{ }</code>	Text	Typewriter / code text.

Table 2.6 L^AT_EX text formatting (series).

Command	Example	Description
<code>\textbf{ }</code>	Text	Bold text. – Shortcut: <code>Ctrl</code> + <code>B</code>
<code>\textmd{ }</code>	Text	Medium text.

Table 2.7 L^AT_EX text formatting (shape).

Command	Example	Description
<code>\textit{ }</code>	<i>Text</i>	Italic text. – Shortcut: <code>Ctrl</code> + <code>I</code>
<code>\textsl{ }</code>	<i>Text</i>	Slanted text.
<code>\textup{ }</code>	Text	Upright text.
<code>\textsc{ }</code>	TEXT	Small uppercase text.

Table 2.8 L^AT_EX text formatting (other).

Command	Example	Description
<code>\emph{ }</code>	<i>Text</i>	Emphasized text. This command is context aware: if used in normal font text, it will default to italic, and vice-versa.

Command	Example	Description
<code>\underline{}</code>	<u>Text</u>	Underlined text. <i>Not recommended</i> : the command encloses its argument in a box, which doesn't allow line breaks.
<code>\textsubscript{}</code>	Text	Subscript inside text.
<code>\textsuperscript{}</code>	Text	Superscript inside text.

2.3.2 SIZE

You can also control the size of your text. The command `\normalsize` only needs to be used to bring back the standard font size inside an environment that uses a smaller size (e.g., a footnote). With some fonts, the text can seem of a different size depending on the family: though this is not the case in reality, you can adjust it by using `{\small ...}` on the seemingly bigger font family. Just make sure it's consistent throughout the document. Last but not least, it is possible to specify a custom size with `\fontsize{ }{\ }\selectfont` inside a group, where the first argument defines the font size, and the second the line skip size, both written as plain numbers. Line skip usually is the closest integer to 120 % the font size.

Table 2.9 L^AT_EX text size commands.

Command	Example
<code>\miniscule</code>	Supported by the <i>memoir</i> document class
<code>\tiny</code>	Text
<code>\scriptsize</code>	Text
<code>\footnotesize</code>	Text
<code>\small</code>	Text
<code>\normalsize</code>	Text
<code>\large</code>	Text
<code>\Large</code>	Text
<code>\LARGE</code>	Text

Command	Example
<code>\huge</code>	Text
<code>\Huge</code>	Text
<code>\HUGE</code>	Supported by the <i>memoir</i> document class

Table 2.10 L^AT_EX text size (article, report, book, letter).

Command	Size (pt)		
<code>\tiny</code>	5	6	6
<code>\scriptsize</code>	7	8	8
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	11
<code>\normalsize</code>	10	11	12
<code>\large</code>	12	12	14
<code>\Large</code>	14	14	17
<code>\LARGE</code>	17	17	20
<code>\huge</code>	20	20	25
<code>\Huge</code>	25	25	25

Table 2.11 L^AT_EX text size (memoir).

Command	Size (pt)											
<code>\miniscule</code>	4	5	6	7	8	9	10	11	12	14	17	20
<code>\tiny</code>	5	6	7	8	9	10	11	12	14	17	20	25
<code>\scriptsize</code>	6	7	8	9	10	11	12	14	17	20	25	30
<code>\footnotesize</code>	7	8	9	10	11	12	14	17	20	25	30	36

Command	Size (pt)											
<code>\small</code>	8	9	10	11	12	14	17	20	25	30	36	48
<code>\normalsize</code>	9	10	11	12	14	17	20	25	30	36	48	60
<code>\large</code>	10	11	12	14	17	20	25	30	36	48	60	72
<code>\Large</code>	11	12	14	17	20	25	30	36	48	60	72	84
<code>\LARGE</code>	12	14	17	20	25	30	36	48	60	72	84	96
<code>\huge</code>	14	17	20	25	30	36	48	60	72	84	96	108
<code>\Huge</code>	17	20	25	30	36	48	60	72	84	96	108	120
<code>\HUGE</code>	20	25	30	36	48	60	72	84	96	108	120	132

Table 2.12 \LaTeX text size (beamer).

Command	Size (pt)							
<code>\tiny</code>	5	5	5	6	6	6	8	10
<code>\scriptsize</code>	5	6	7	8	8	8	10	12
<code>\footnotesize</code>	6	7	8	9	10	10	12	14
<code>\small</code>	7	8	9	10	11	12	14	17
<code>\normalsize</code>	8	9	10	11	12	14	17	20
<code>\large</code>	10	10	12	12	14	17	20	25
<code>\Large</code>	11	11	14	14	17	20	25	29.86
<code>\LARGE</code>	12	12	17	17	20	25	29.86	35.83
<code>\huge</code>	14	14	20	20	25	29.86	35.83	42.99
<code>\Huge</code>	17	17	25	25	25	35.83	42.99	51.59

2.3.3 SPACING

Most of the time, spacing is intuitive and simple: hit the space bar, just like you would in any other typesetting software. But, in some situations, such as a mathematical environment or a weird combination of font formatting, you might want to get granular and fine-tune your spacing.

Tables 2.13 and 2.14 list the most common commands to do so.

Some of these commands possess a useful although strange behavior: they exhibit compression or expansion. To understand this behavior, we need to define the skip. A skip is a three-pronged length specification processed by the \TeX engine with the following syntax:

`<natural width> plus <stretch> minus <shrink>`

For example, `\smallskip` equates to `\vspace{3pt plus 1pt minus 1pt}` for standard document classes. That tells \TeX to leave a vertical space of 3 pt, but shrinkable up to 2 pt or stretchable up to 4 pt. However, when a stretch component is present, \TeX is allowed to stretch that space beyond the stated specification in an emergency: this happens frequently when a `\pagebreak` command is found when \TeX has not enough material to fill the current page correctly.

The lengths can be specified in any of the legal \TeX dimensions, but the stretch and shrink components can be expressed also in terms of infinite units:

- `fil`: first order infinity;
- `fill`: second order infinity;
- `filll`: third order infinity.

Normally, `fil` is reserved for standard macros, and `fill` is for user macros that override the previous order infinities. `filll` is seldom used and should be reserved for emergency cases. Continuing with our example, if \TeX finds `\vfill`—same as `\vspace{0pt plus 1fill}`—in a page, uses of `\smallskip` found in the page will be at most 3 pt wide, as the infinite component “wins”.

Table 2.13 \LaTeX horizontal spacing commands.

Command	Description
<code>\,</code>	Inserts a space of 0.16667 em in text mode.
<code>\thinspace</code>	Equivalent to <code>\,</code> .
<code>\:</code>	Inserts a space of 0.22223 em in text mode.
<code>\></code>	Equivalent to <code>\:</code> .
<code>\medspace</code>	Equivalent to <code>\:</code> .

Command	Description
<code>\;</code>	Inserts a space of 0.27777 em in text mode.
<code>\thickspace</code>	Equivalent to <code>\;</code> .
<code>\!</code>	Negative equivalent to <code>\,</code> .
<code>\negthinspace</code>	Equivalent to <code>\!</code> .
<code>\negmedspace</code>	Negative equivalent to <code>\:</code> .
<code>\negthickspace</code>	Negative equivalent to <code>\;</code> .
<code>_</code>	Control space. The <code>_</code> character only highlights the necessary empty space.
<code>\space</code>	Equivalent to normal spacing.
<code>~</code>	Non-breakable space.
<code>\thinmuskip</code>	Inserts a space of 3 mu in math mode.
<code>\medmuskip</code>	Inserts a space of 4 mu, shrinkable to 0 mu and stretchable to 6 mu, in math mode.
<code>\thickmuskip</code>	Inserts a space of 5 mu, stretchable to 10 mu, in math mode.
<code>\enskip</code>	Inserts a space of 0.5 em.
<code>\enspace</code>	Equivalent to <code>\enskip</code> .
<code>\quad</code>	Inserts a space of 1 em.
<code>\qqquad</code>	Inserts a space of 2 em.
<code>\kern</code>	Plain \TeX skip. Must be followed by a space and its argument. Inserts a space the size of the dimension in the parameter. <i>Usage not recommended.</i>
<code>\hskip</code>	Equivalent to <code>\kern</code> . <i>Usage not recommended.</i>
<code>\hspace{}</code>	\LaTeX skip. Inserts a space the size of the dimension in the parameter. Use the variant <code>\hspace*{}</code> to make sure the spacing is not removed if it comes at the end of a line.

Command	Description
<code>\hfill</code>	Inserts a stretchable spacing between elements. If result is unsatisfactory, a suitable replacement is <code>\hspace*{\hfill}</code> .
<code></code>	Inserts a box with the width and height of the argument. The parameter can be any value that takes a non-zero amount of space when printed in the document.
<code>\hphantom{ }</code>	Variant of <code></code> where the height is equal to zero.
<code>\hss</code>	Horizontal glue that is both infinitely stretchable and shrinkable. <i>Handle with care.</i>

Table 2.14 \LaTeX vertical spacing commands.

Command	Description
<code>\smallskip</code>	Inserts a space of 3 pt, shrinkable to 2 pt and stretchable to 4 pt.
<code>\medskip</code>	Inserts a space of 6 pt, shrinkable to 4 pt and stretchable to 8 pt.
<code>\bigskip</code>	Inserts a space of 12 pt, shrinkable to 8 pt and stretchable to 16 pt.
<code>\smallbreak</code>	Variant of <code>\smallskip</code> that also removes a preceding vertical space if it is less than what they would insert.
<code>\medbreak</code>	Variant of <code>\medskip</code> that also removes a preceding vertical space if it is less than what they would insert.
<code>\bigbreak</code>	Variant of <code>\bigskip</code> that also removes a preceding vertical space if it is less than what they would insert.
<code>\baselineskip</code>	Distance between the bottom of two lines of text inside a paragraph.
<code>\vskip</code>	Plain \TeX skip. Must be followed by a space and its argument. Inserts a space the size of the dimension in the parameter. <i>Usage not recommended.</i>
<code>\vspace{ }</code>	\LaTeX skip. Inserts a space the size of the dimension in the parameter. Use the variant <code>\vspace*{ }</code> to make sure the spacing is not removed if it comes at the end of a page.

Command	Description
<code>\addvspace{ }</code>	Variant of <code>\vspace{ }</code> that also removes a preceding vertical space if it is less than what they would insert.
<code>\vfill</code>	Inserts a stretchable spacing between elements. If result is unsatisfactory, a suitable replacement is <code>\vspace*{\fill}</code> .
<code></code>	Inserts a box with the width and height of the argument. The parameter can be any value that takes a non-zero amount of space when printed in the document.
<code>\vphantom{ }</code>	Variant of <code></code> where the width is equal to zero.
<code>\vss</code>	Vertical glue that is both infinitely stretchable and shrinkable. <i>Handle with care.</i>

2.4 SECTIONS

\LaTeX defines many different levels of document organization. Following the guidelines of the *Protocole de rédaction aux études supérieures* provided by the Université de Sherbrooke [8], the project uses three levels only in my documents: the section, the subsection, and the sub-subsection. In your own documents, you can use whatever structure you see fit, just make sure to update the labelling scheme so it's coherent between different levels (see § 5.20).

Table 2.15 \LaTeX section levels.

Level	Command	Description
-1	<code>\part{ }</code>	Supported by the <i>book</i> and <i>report</i> document classes
0	<code>\chapter{ }</code>	Supported by the <i>book</i> and <i>report</i> document classes
1	<code>\section{ }</code>	First level for most document classes
2	<code>\subsection{ }</code>	Second level for most document classes
3	<code>\subsubsection{ }</code>	Third level for most document classes
4	<code>\paragraph{ }</code>	Fourth level for most document classes
5	<code>\subparagraph{ }</code>	Fifth level for most document classes

2.5 LINKS AND REFERENCES

In \LaTeX , you can easily reference almost anything that can be bookmarked, and have it automatically updating the marker for you whenever necessary. You can also build your bibliography with a \BibTeX database, or add hyperlinks throughout your document. The following sections will cover them separately.

2.5.1 LABELLING

Labelling and referencing is very easy. It only requires two commands: `\label{}` and `\ref{}`. The former creates an argument that can be used by the latter to create a hyperlink associated with a specific floating environment instance, or object, at a specific spot in the document. Most templates (see § 6) add a label where most appropriate, except for lists: only ordered lists support labelling properly, and their appeal is limited since only items can be labelled and not the list itself. Glossary entries, acronyms and sources have their own independent, integrated approach: the two commands above are unnecessary.

Since the same commands can be used to reference almost anything, it might get a bit confusing when a lot of references are introduced. Two mitigation strategies can be used to make it easier to find the right one:

- add a few letters to the label to describe what you are referencing (see table 2.16). It’s common practice among \LaTeX users. Some packages, such as `fancyref`, rely on this metadata;
- add your directory inside the label. Labels can get quite lengthy with that method, but it’s easier to find when looking at the document structure and using the auto-completion feature of Overleaf.

Examples of labels can be seen in table 2.16. Note that, past the colon (:), words like “doc”, “sec”, “subsec”, and other refer to the name of the document, the section, the subsection, etc., inside the project architecture. For instance, if the document folder is named “tutorial”, you can replace “doc” with it in all your labels inside that document (more information about the project architecture in § 4).

Table 2.16 L^AT_EX labelling examples.

Type		Examples
Section	sec:	<ul style="list-style-type: none"> – <code>\label{sec:doc/sec}</code> – <code>\label{sec:doc/sec/subsec}</code> – <code>\label{sec:doc/sec/subsec/subsubsec}</code> – ...
Figure	fig:	<ul style="list-style-type: none"> – <code>\label{fig:doc/sec/fig}</code> – <code>\label{fig:doc/sec/subsec/fig/subfig}</code> – ...
Table	tab:	<ul style="list-style-type: none"> – <code>\label{tab:doc/sec/tab}</code> – <code>\label{tab:doc/sec/subsec/tab/subtab}</code> – ...
Equation	eq:	<ul style="list-style-type: none"> – <code>\label{eq:doc/sec/equation}</code> – <code>\label{eq:doc/sec/subsec/equation}</code> – ...
Code listing	lst:	<ul style="list-style-type: none"> – <code>\label{lst:doc/sec/code}</code> – <code>\label{lst:doc/sec/subsec/code}</code> – ...

2.5.2 HYPERLINKS

Hyperlinks can be added anywhere in the document with a handful of commands:

- `\url{}`: adds a uniform resource locator (URL) inside the text;
- `\href{}{}`: adds a URL inside the document (first argument) and replaces it with text (second argument);
- `\hyperref[]{}{}`: refers to a label (first argument) and replaces it with text (second argument);
- `\hyperlink{}{}`: refers to a label (first argument) and replaces it with text (second argument);
- `\hypertarget{}{}`: creates a label (first argument), writes text (second argument), and adds its position in the document as a reference point.

2.5.3 BIBLIOGRAPHY

`BIBTEX` and `biber` are reference management software for processing bibliographical information and act as the interface between your `.bib` file (see § 1.5) and your `LATEX` document. It makes it easy to cite sources in a consistent manner, by separating the data from the style, similarly to the separation of content and presentation/style supported by `LATEX` itself [9].

`BIBTEX` / `biber` and their format layer `BIBLATEX` split sources by type (book, article, web, etc.). Then, within a source, the information is split by field (author, date, URL, etc.). For every type of source covered by the *Protocole de rédaction aux études supérieures* of the Université de Sherbrooke [8], there is a template available in `templates/template.bib` in the `src/` folder (as seen in § 6.6). There are two packages for bibliography formatting, `biblatex` and `natbib`, the former being the better choice since it supports the more powerful backend `biber`.

- `\cite[] [] {}`: adds a citation inside the text. The first argument defines a note before the citation, the second a note after, and the third is the label.
- `\parencite[] [] {}`: adds a citation inside the text between parentheses or brackets. The first argument defines a note before the citation, the second a note after, and the third is the label. We will favor this method, as custom macros have been set to modify it when using the “author-date” citationciting style. Using this command will allow the writer to switch between the “numeric” and the “author-date” citationciting styles without any modifications other than in the preamble (see § 5.4);
- `\citename{} {}`: includes the value of the specified source field. The first argument is the label, and the second one is the field. If this command doesn’t work because `BIBTEX` doesn’t consider some fields as name fields, you can also try `\citelist{} {}` or `\citefield{} {}`.

For some sources, such as websites or web documents, it is sometimes customary to include them as footnotes. We can use the command `\footcite[] [] {}`, which works the same way as `\cite[] [] {}` and `\parencite[] [] {}`. If we want more control over the styling, we can use something like this:

```
\footnote{\citeauthor{}.\sim\cite{}}
```

This will put a page footnote using the author and the URL of a website or web document. The citing commands only need the label.

For more options and commands, feel free to consult the `BIBLATEX` cheat sheet or the CTAN documentation. [10, 11]

2.5.4 GLOSSARY

Glossary entries, just like sources, are kept in a database to separate the information from the styling. We can refer to them with a handful of commands:

- `\gls{}`: refers to a glossary entry;
- `\Gls{}`: refers to a glossary entry and capitalize the first letter;
- `\glspl{}`: refers to a glossary entry in plural form. It defaults to adding the letter “s”. For complex entries, it is advised to define it yourself;
- `\Glspl{}`: refers to a glossary entry in plural form and capitalize the first letter;
- `\glsname{}`: refers to only the name of a glossary entry;
- `\Glsname{}`: refers to only the name of a glossary entry and capitalize the first letter;
- `\glsdesc{}`: refers to only the description of a glossary entry;
- `\Glsdesc{}`: refers to only the description of a glossary entry and capitalize the first letter;
- `\glsdisp{}{}`: refers to a glossary entry and change the displayed text (useful for conjugation). The first argument is the label, and the second is the text shown in the document.

It is recommended to use `\gls{}` or `\glspl{}` and their capitalized variants since these will automatically modify the first occurrence in your text (especially useful for acronyms—see § 2.5.5). Choose `\glsname{}` or `\glsdesc{}` and their capitalized variants if you need to manually control the output (table header, figure caption, etc.).

All the above commands can create hyperlinks to your glossary entry, which helps navigate the document to find occurrences and guide the reader to the description of a technical term, for instance. However, we do not always need a hyperlink. We may need to avoid it entirely if it’s included in a hyperlink already, e.g., a section title: if we leave the glossary link in, part of the title in the ToC will not refer to the section, but instead to the glossary. To prevent this in those edge cases, we can use the following commands:

- `\glsentrytext{}`: refers to only the name of a glossary entry without hyperlink functionality;
- `\Glsentrytext{}`: refers to only the name of a glossary entry without hyperlink functionality and capitalize the first letter;
- `\glsentrydesc{}`: refers to only the description of a glossary entry without hyperlink

functionality;

- `\glsentrydesc{}`: refers to only the description of a glossary entry without hyperlink functionality and capitalize the first letter.

When defining your glossary entry, you only have to define a couple fields inside the entry (example at § 6.7):

- `name={}`: the word you're adding in the glossary;
- `description={}`: introduces the reader to the meaning of your entry;
- `see=[]{}`: refers to another glossary entry if needed. If you're not a fan of the command's behavior, you can also refer to another entry by using `\glsseelist{}` in the entry's description.

For more options and macros, feel free to consult the CTAN documentation [12].

2.5.5 ACRONYMS

Acronyms can be referred to with the same commands as a glossary entry since they're both defined by the same package. They simply use more fields to add some features that are not or seldom needed with glossary entries:

- `type={}`: the value for acronyms is `\acronymtype`. This allows us to create separate indexes for glossary entries and acronyms;
- `first={}`: defines the output when the acronym is first referred to;
- `plural={}`: defines the plural form of the acronym;
- `firstplural={}`: defines the output when the acronym is first referred to in plural form.

Most acronyms can use `\glsentrytext{}` to refer to the name when redefining the first or plural form, and `\glsentrydesc{}` to refer to the description (see § 2.5.4). Some examples can be seen in § 6.8.

2.6 LIST

Lists are a very simple environment. You only need to know three commands:

- `item`, to declare the elements of a list;
- `itemize`, the environment for unordered lists;
- `enumerate`, the environment for ordered lists.

That’s all there is! If you need a list inside a list, you can nest your environment instances as needed. In this project, bullets are defined for up to four levels. If you need more or you want to redefine the bullets, you can do so in the preamble (see § 5.21). For list examples, refer to § 6.2.

2.7 MATH

Accurate representation of mathematics was one of the main reasons for the creation of \LaTeX . Once you know how to tweak equations, their appearance will be stellar. For more documentation and examples on equations, be sure to checkout Herbert Voß’s guide on mathematical typesetting and review of mathematics [13, 14].

2.7.1 ENVIRONMENTS

Mathematical equations can be included in your document with a variety of environments. Table 2.17 lists the common ones.

Note: Some math environments will allow automatic numbering of equations by default, such as `equation`, `align`, and `eqnarray`. To prevent this behavior, you can declare them using an asterisk:

```
\begin{align*}...\end{align*}
```

Maybe you want to number the last line only. In order to do this, you will have to use a numbering environment and declare `\nonumber` on every line that shouldn’t have a number.

You can also allow referencing of your equations or equation sets by adding a label before closing the math environment (see § 2.5.1 for labelling scheme and § 6.1 for examples).

Table 2.17 \LaTeX mathematical environments.

Environment	Description
<code>math</code>	Inline equations (inside text). – Shortcut: <code>\$...\$</code>
<code>displaymath</code>	Unnumbered equations that stand on their own line. – Shortcut: <code>\[...\]</code>
<code>equation</code>	Numbered equations that stand on their own line.

Environment	Description
<code>align</code>	Two-column environment that can be used to align equations (alignment character: <code>&</code>). The first column is right-aligned, the second is left-aligned.
<code>eqnarray</code>	Three-column environment that can be used to align equations (alignment character: <code>&</code>). The first column is right-aligned, the second is centered, and the third is left-aligned.
<code>matrix</code>	<p>Matrices (alignment character: <code>&</code>; row change: <code>\\</code>).</p> <ul style="list-style-type: none"> – <code>pmatrix</code>: matrix with parentheses; – <code>bmatrix</code>: matrix with brackets; – <code>Bmatrix</code>: matrix with braces; – <code>vmatrix</code>: matrix with vertical bars; – <code>Vmatrix</code>: matrix with double vertical bars.

2.7.2 MACROS

A lot of special commands are required for proper mathematical formulas. Table 2.18 lists the most common and interesting ones. A full list of math symbols can be found on Matthias Heinkenschloss’s portal at Rice University [15].

Table 2.18 \LaTeX mathematical commands.

Command	Examples	Description
<code>\mathcal{}</code> <code>\mathfrak{}</code> <code>\mathbb{}</code>	\mathcal{L} \mathfrak{R} \mathbb{O}	Math fonts for special characters (Calligraphic, Fraktur, Blackboard).
<code>\mathsf{}</code> <code>\mathtt{}</code>	ABC ABC	Math families (sans serif and typewriter).
<code>\mathbf{}</code>	ABC	<p>Math series (bold).</p> <ul style="list-style-type: none"> – May not work with every font; – Conflicts with package <code>newtxmath</code>.

Command	Examples	Description
<code>\mathit{}</code>	<i>ABC</i>	Math shape (italic). – May not work with every font; – Conflicts with package <code>newtxmath</code> .
<code>\mathrm{}</code> <code>\text{}</code>	ABC ABC	Remove math environment. – <code>\mathrm{}</code> for symbols; – <code>\text{}</code> for text.
<code>\frac{ }{ }</code> <code>\cfrac{ }{ }</code> <code>\tfrac{ }{ }</code>	$\frac{A}{B}$	Fraction. – <code>\cfrac{ }{ }</code> : constant height fraction; – <code>\tfrac{ }{ }</code> : text fraction.
<code>\int</code> <code>\iint</code> <code>\iiint</code> <code>\idotsint</code>	\int \iint \iiint $\int \cdots \int$	Simple, double, triple, and n integrals.
<code>\oint</code> <code>\oiint</code> <code>\varoiint</code>	\oint \oiint \varoiint	Path and surface integrals.
<code>\sum</code> <code>\prod</code> <code>\coprod</code>	Σ Π \coprod	Sum, product, and coproduct.
<code>\cos</code> <code>\sin</code> <code>\tan</code>	cos sin tan	Trigonometric functions. – For adjustable parentheses after the function, use the custom macro in § 2.12.
<code>\alpha</code> <code>\beta</code> ...	α β ...	Greek letters.
<code>\limits_{}^{\{}}</code>	\int_0^N $\sum_{n=0}$	Subscripts and superscripts under and over a symbol.

2.8 FIGURE

Figures in \LaTeX are declared in an environment that requires what we call a “float specifier”. Floats are containers for things in a document that cannot be broken over a page. Tables and figures are considered floats. We can alleviate the issue for table with the `longtable` package (see § 2.9). For figures though, we need a float specifier as a second argument when beginning the environment (`\begin{figure}[]`). Table 2.19 lists the standard ones, which can be combined together if necessary. In general, I recommend using `H`.

Table 2.19 \LaTeX float specifiers.

Symbol	Description
<code>h</code>	Place the float <i>here</i> , i.e., <i>approximately</i> at the same point it occurs in the source text (however, not <i>exactly</i> at the spot).
<code>t</code>	Position at the top of the page.
<code>b</code>	Position at the bottom of the page.
<code>p</code>	Put on a special page for floats only.
<code>!</code>	Override internal parameters \LaTeX uses for determining good float positions.
<code>H</code>	Places the float at precisely the location in the \LaTeX code (requires the <code>float</code> package).

Figures can contain subfigures, declared with the `subfigure` environment. They will need to be aligned with each other. To do this, we use an alignment specifier as a second argument when beginning a subfigure, the same way we specify the float type for a figure. Table 2.20 contains the standard ones.

Table 2.20 \LaTeX subfigure alignment specifiers.

Symbol	Description
<code>b</code>	Align subfigures at the bottom.
<code>c</code>	Align subfigures at the center.

Symbol	Description
<code>t</code>	Align subfigures at the top.

To create a figure, we need a handful of macros:

- `\centering`: centers the figure;
- `\includegraphics[]{}:` includes an image inside the figure. In the first argument, we can define the field `width` to make sure the figure stays within the page inside the text area. Other fields can be found in the `graphicx` package documentation [16]. The second argument is the image path, name and extension, e.g. `data/.../image.pdf` (see § 4.3). We could also include a drawing using the progressive graphics file (PGF) format with the `tikz` package instead (see § 6.5);
- `\columnwidth`: defines the width of a column of text. This can be used to define the `width` field of `\includegraphics[]{}:` I also recommend appending a custom command, which can be any value between 0 and 1, to modulate the figure width globally. I have created the command `\figsize` just for that (see § 2.12);
- `\textwidth`: defines the width of the text area. Use this one with a ratio between 0 and 1 to specify the subfigure environment width;
- `\caption[]{}:` defines the name of the figure. The first argument is the name as seen in a LoF, the second is the one displayed over or under the figure;
- `\label{}:` see § 2.5.1.

For some examples, visit § 6.3.

2.9 TABLE

Tables in \LaTeX are, in my opinion, the most complex environment to setup. Tables benefit extensively from the WYSIWYG system: it’s easy to draw a table the way you want it to look in Microsoft Word for instance. Not so in \LaTeX .

2.9.1 ENVIRONMENTS

To overcome the difficulty of setting up a table, *many* people created `table` package. The most commonly used ones are listed in table 2.21. I usually recommend using `longtable` so it can easily break over to the next page if necessary, though it doesn’t reckon well with multi-column

documents or stretching columns—columns that make sure the table is as wide as the text area. I am currently looking for a replacement that could alleviate those issues: the table sections in this document are subject to change.

A word of warning: Adding another environment, such as `itemize`, inside a table is *not recommended*, as it’s not aware of a table’s functions and variables. These environments define everything for themselves, and therefore are not compatible with other environments, except where compatibility was explicitly enabled (`figure` with `subfigure`, `equation` with `align`, etc.). To replace the functionality of a list, we can define a bullet point with the custom command `\tabitem` (see § 2.12). Some examples are available in § 6.4.

For more documentation and examples about all types of tables, be sure to checkout Herbert Voß’s guide on table typesetting [17]. If you’re not a fan of the “prison” style of tables—which works very well for reports in my opinion, you can checkout Markus Püschel’s guide on nice tables [18].

Table 2.21 \LaTeX table environments.

Environment	Description
<code>table</code>	Floating environment container for tables. Requires float specifier (see table 2.19).
<code>tabular</code>	Default \LaTeX table environment.
<code>tabularx</code>	Tables with adjustable-width columns.
<code>supertabular</code>	A multi-page tables package.
<code>longtable</code>	Allow tables to flow over page boundaries. Doesn’t support stretching (X columns) or multi-column documents.
<code>tabu</code>	Flexible \LaTeX tabulars.

2.9.2 COLUMNS

When beginning a table environment, one of the first arguments we must declare is the structure. This can be comprised of any number of columns, of any type. Table 2.22 lists the standard column types. Columns in the declared structure can be split with some characters, such as `|` for vertical bars between them, or `||` for double vertical bars.

Table 2.22 \LaTeX table columns.

Column	Description
<code>l</code>	Left-justified column.
<code>c</code>	Centered column.
<code>r</code>	Right-justified column.
<code>p{}</code>	Paragraph column with text vertically aligned at the top. Width must be specified.
<code>m{}</code>	Paragraph column with text vertically aligned in the middle. Width must be specified. Requires the <code>array</code> package (§ 5.16).
<code>b{}</code>	Paragraph column with text vertically aligned at the bottom. Width must be specified. Requires the <code>array</code> package (§ 5.16).

2.9.3 MACROS

Tables are definitely the environment that requires the most commands to create and customize:

- `\arraystretch`: specifies the row padding. I usually set it the same as the line spacing, which is defined by `\baselinestretch` and can be modified with `\linespread{}`. To do this, you need to renew the command:

```
\renewcommand{\arraystretch}{\baselinestretch}
```

You can either do this for every table if you want different values, or set it globally in the preamble (see § 5.16);

- `\tabcolsep`: specifies the column padding. I recommend setting this value according to the font size selected for the table (see § 2.3.2), so the proportions stay consistent between tables. You can set the value with `\setlength{\tabcolsep}{}`;
- `\linewidth`: defines the width of a line in the local environment. I use this length and a ratio between 0 and 1 to specify the column width of each column;
- `\hline`: draws a horizontal line. Before using it, you must use `\\` to change row inside the table;
- `\cline{}`: draws a horizontal line only between the specified columns. Before using it, you

must use `\\` to change row inside the table. The range should be specified with two numbers separated by a hyphen;

- `\newline`: allows you to start a new line within a cell;
- `&`, the column separator. It allows you to switch columns when adding content to your table;
- `\endfirsthead`: defines, in the `longtable` environment, everything above it and below the last environment-specific macro to be the first header of the table. When the table overflows to the next page, that header will not appear. It will be necessary to specify a separate header for the first page for captions and labels on top: else, they will be repeated on every page and create warnings;
- `\endhead`: defines, in the `longtable` environment, everything above it and below the last environment-specific macro to be the default header of the table. When the table overflows to the next page, this header will be repeated. It will also act as the first header of the table unless `\endfirsthead` is used;
- `\multicolumn{}{}{}{}:` allows us to define multi-column content. The first argument is the number of columns to cover, the second is the structure of that new multi-column (type and vertical bars if needed), and the third is the content;
- `\multirow{}{}{}{}:` allows us to define multi-row content. The first argument is the number of rows to cover, the second is the width (use an asterisk for the “natural” width of those cells, though it doesn’t allow wrapping; else, use the width of that column), and the third is the content. Vertical alignment is not supported;
- `\caption[]{}{}:` defines the name of the table. The first argument is the name as seen in a list of tables, the second is the one displayed over or under the table;
- `\label{}{}:` see § 2.5.1.

2.10 TYPESETTING

The `lstlisting` environment is a kind of extensible / configurable typesetting environment, much like a supercharged version of `verbatim`:

- it displays code as-is;
- it provides labelling of code chunks with the `\caption{[]}` / `\label{}{}:` (see § 2.5.1). Notice the difference between the caption command in § 2.8 or § 2.9.3 and this one: the name for the list of listings must be declared inside the braces, with the name under the listing;
- it provides syntax highlighting for a variety of programming languages. One can even set their

own language.

The environment is set with `\begin{lstlisting}[]`. Options such as `language`, `caption`, and `label` are set between the brackets.

```
1 \begin{lstlisting}[
2   language = LaTeX,
3   caption  = {\LaTeX{} listing \glsentrytext{environment}}\LaTeX{} listing \gls{environment}
4   label    = {lst:tutorial/latex/listing}
5 ]
6 ...
7 \end{lstlisting}
```

Listing 2.1 \LaTeX listing environment.

2.11 GRAPHICS

There are two main ways to draw graphics directly in your documents: PGF/TikZ or PSTricks. The former is better maintained, easier to learn, and more extendable. The latter is more powerful, but it has a steep learning curve. We will stick with TikZ for these reasons. For more documentation and examples on the PSTricks packages, be sure to checkout Herbert Voß’s guide on PostScript graphics [19]. Readers beware: it’s a great resource that is unfortunately riddled with mistakes.

The use case for TikZ can be so specific, I won’t bother trying to list all the different options and macros. Rather, I will point readers towards the \LaTeX Wikibooks page—which is an easy way to start—and the CTAN documentation [20, 21].

2.12 CUSTOM MACROS

It can often be useful to create your own commands to simplify usage or provide another layer of abstraction for quick and dirty modifications of a document. Some of the most important ones I created are:

- `\guil{}:` adds quotation marks. This layer of abstraction allows us to redefine the command depending on the document language (“ ” for English, « » for French, etc.);
- `\inp{}:` parentheses that adjust to the content’s height in math mode. Improves the aesthetics of equations while being simpler to implement than the underlying code;

- `\insb{}`: brackets that adjust to the content’s height in math mode. Improves the aesthetics of equations while being simpler to implement than the underlying code;
- `\incb{}`: braces that adjust to the content’s height in math mode. Improves the aesthetics of equations while being simpler to implement than the underlying code;
- `\sfrac{ }{ }`: a custom slanted fraction that doesn’t rely on specialty packages;
- `\figsize`: defines a ratio between 0 and 1 to scale figures;
- `\tabitem`: defines a bullet for lists inside float environments that don’t support `itemize` or `enumerate`.

2.12.1 TABLE COLUMNS

Standard column types have a pretty limited usage: they either don’t allow text wrapping and/or don’t provide vertical alignment and/or don’t provide horizontal alignment. That’s why I created nine new column types, that provide every type of vertical or horizontal alignment while all supporting text wrapping. They all have width as an argument.

Table 2.23 Custom table columns.

Command	Description
<code>\LT{ }</code>	Left-aligned horizontally, top-aligned vertically.
<code>\CT{ }</code>	Centered horizontally, top-aligned vertically.
<code>\RT{ }</code>	Right-aligned horizontally, top-aligned vertically.
<code>\LC{ }</code>	Left-aligned horizontally, centered vertically.
<code>\CC{ }</code>	Centered horizontally, centered vertically.
<code>\RC{ }</code>	Right-aligned horizontally, centered vertically.
<code>\LB{ }</code>	Left-aligned horizontally, bottom-aligned vertically.
<code>\CB{ }</code>	Centered horizontally, bottom-aligned vertically.
<code>\RB{ }</code>	Right-aligned horizontally, bottom-aligned vertically.

Column width can also be finicky. For instance, if we want the table to be *exactly* the width of the text area, we need to take the table column separation and the vertical rule thickness into account. These values can all be expressed as a ratio of `\linewidth`, $k \in \{0, 1\}$:

$$1 = \sum_{n=1}^N k_n + 2Nk_{\text{tabcolsep}} + M_{|}k_{|} + M_{||}k_{||} \quad (2.1)$$

- k_n is the thickness of each column;
- $k_{\text{tabcolsep}}$ is the thickness of the tabular separation between columns. It appears twice between columns and once on each side of the table;
- $k_{|}$ is the thickness of a singular vertical bar, defined as `\vbar`;
- $k_{||}$ is the thickness of a double vertical bar, defined as `\doublevbar`;
- $M_{|}$ is the number of single vertical bars;
- $M_{||}$ is the number of double vertical bars;
- N is the number of columns.

For our next table, we must redefine columns so the sum of all elements in equation 2.1 is always equal to `\linewidth`. To do this, we can first start splitting our columns as if table column separation and vertical bars didn't exist, like so:

```

1 \setlength{\columnA}{16\linewidth\31}
2 \setlength{\columnB}{8\linewidth\31}
3 \setlength{\columnC}{4\linewidth\31}
4 \setlength{\columnD}{2\linewidth\31}
5 \setlength{\columnE}{\linewidth-\columnA-\columnB-\columnC-\columnD}

```

Listing 2.2 Column length assignment (I).

Then, we need to account for the dimensions we just ignored. Every column as padding on both sides, so this one is easy. Vertical bars are trickier: assuming we separate every column with vertical bars, and our table has a frame, we have one more bar than there are columns. We can then split our total bar thickness by the number of columns for each column. If there were a mix of single and double vertical bars, same principle applies: divide the number of single bars by the number of columns for each, and the number of double bars by the numbers of columns for each:

```

1 \setlength{\columnA}{\columnA-2\tabcolsep-6\vbar/5-0\doublevbar/5}
2 \setlength{\columnB}{\columnB-2\tabcolsep-6\vbar/5-0\doublevbar/5}
3 \setlength{\columnC}{\columnC-2\tabcolsep-6\vbar/5-0\doublevbar/5}
4 \setlength{\columnD}{\columnD-2\tabcolsep-6\vbar/5-0\doublevbar/5}
5 \setlength{\columnE}{\columnE-2\tabcolsep-6\vbar/5-0\doublevbar/5}

```

Listing 2.3 Column length assignment (II).

If we use `\multicolumn{}{}{}{}` later inside the table, we may not need to specify the width: if we want it to use all the available space, we can specify the default column types `l`, `c`, and `r` (see table 2.22). However, if we need to specify the vertical alignment, we will need to use custom column types (table 2.23) that take width as an argument. To do this, you need to assign it the sum of all column widths + whatever there is in between. Continuing further with our example:

```

1  \begin{longtable}%
2      {|\RC{\columnA}| % Right + center alignment
3       \CC{\columnB}| % Center + center alignment
4       \CC{\columnC}| % Center + center alignment
5       \CC{\columnD}| % Center + center alignment
6       \LC{\columnE}| % Left + center alignment
7      }
8      ...
9
10     % First header
11     \hline
12     \multicolumn{1}{|\CT{\columnA}|}{Column title \#1}
13     & \multicolumn{3}{|\CT{\columnB+\columnC+\columnD+4\tabcolsep+2\vbar}|}{Column title \#2}
14     & \multicolumn{1}{|\CT{\columnE}|}{Column title \#3}
15     \\ \hline
16     \endfirsthead
17     ...

```

Listing 2.4 Multi-column with custom column widths.

2.13 DEBUGGING TOOLS

If you wish to find the length of a variable, some characters, a text excerpt, etc., you can use commands in listing 2.5 to display them in text instead of searching for the value in the compilation logs.

```

1  % Create new length
2  \newlength{\mylength}
3  \newlength{\mywidth}
4  \newlength{\myheight}
5  \newlength{\mydepth}
6
7  % Set length
8  \setlength{\mylength}{...}
9  \settowidth{\mywidth}{...}
10 \settoheight{\myheight}{...}
11 \settodepth{\mydepth}{...}
12

```

```
13 % Display length
14 \the\mylength
15 \the\mywidth
16 \the\myheight
17 \the\mydepth
```

Listing 2.5 Display length in text.

3 **T**_E**X**NICIAN BASICS

COMING IN A FUTURE VERSION

4 ARCHITECTURE

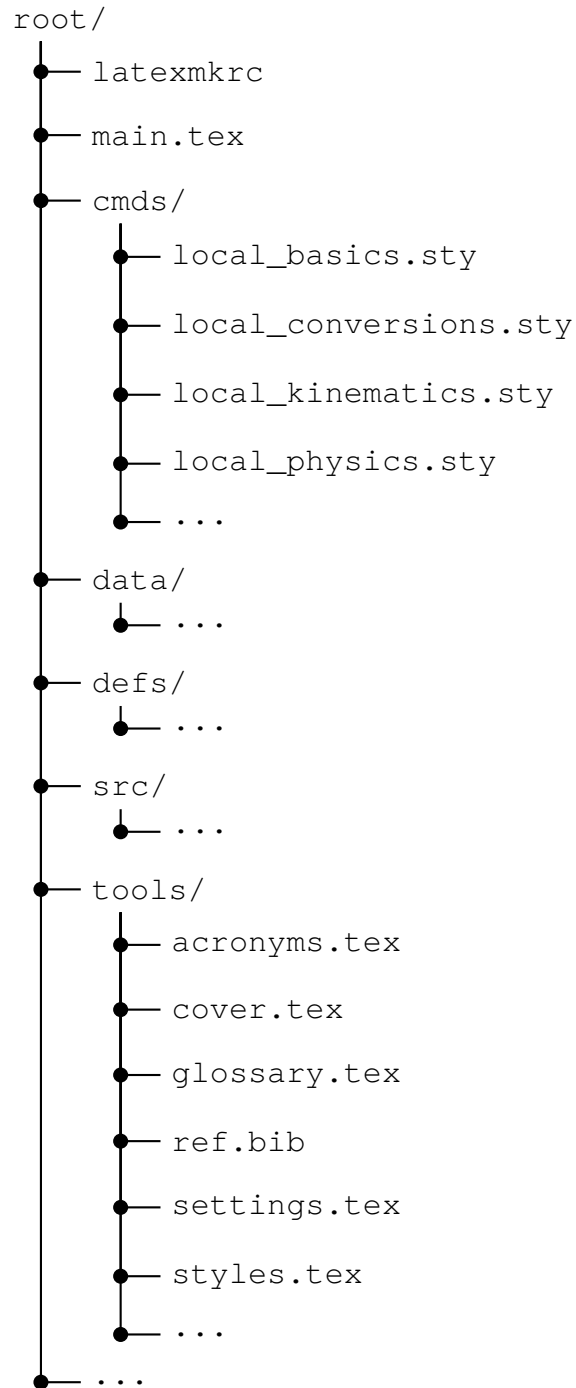


Figure 4.1 Global architecture.

A previous version of the global architecture included a folder for top-level document files, called `comp/`, and a data folder that used to be named `imgs/`. This has been deprecated for future versions of this template. If you end up using the deprecated architecture, just know that the `doc.tex` file in § 4.5 can be found in your `comp/` folder.

4.1 ROOT FILES

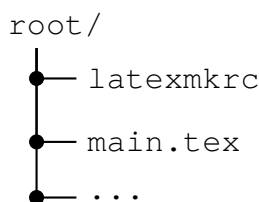


Figure 4.2 Root files.

4.1.1 LATEXMKRC

If you’ve never seen it before, a `latexmkrc` file is a configuration / initialization file for the `Latexmk` package. `Latexmk` is used by Overleaf to control the compilation of your plain text files into the final typeset PDF file. By using a customized configuration file called `Latexmk`, you can override the default compilation commands to allow Overleaf to compile your document in a special way [22].

In our case, I’m using it to specify the default location of custom packages, which is needed for the preamble (see § 4.2 and § 5.23), and the time zone for the `\today` date command in the cover page.

4.1.2 MAIN.TEX

The main file only contains two commands:

- `\RequirePackage{import}`, which makes sure that the `import` package is available to add the content of other `TEX` files;
- an `import` call to select the document to compile, in this case `tutorial.tex` inside the `src/` folder. This allows us to define documents in completely different ways, although we can only compile one at a time.

4.1.3 OTHER OPTIONS

In this project, we have two other files in the root folder:

- `.gitattributes`, which makes sure that every Windows-specific line return (`CR LF`) gets converted to a UNIX-compliant line return (`CR`) for every \TeX file (`.cls`, `.sty`, `.tex`, etc.). This is necessary because the \TeX processor doesn't manage non-UNIX line returns. Binary files are ignored;
- `README.MD`, which describes the project architecture on a GitHub mirror. The project is synced over GitHub to keep track of changes and save a copy in the advent of a major failure or an accidental wiping of files.

4.2 CMDS/

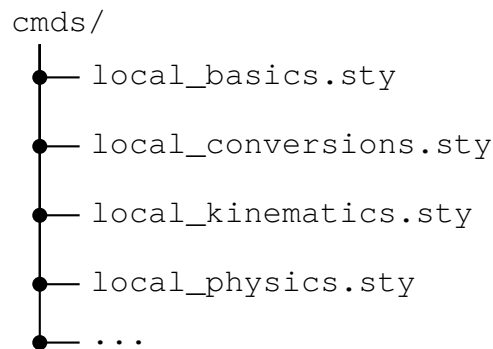


Figure 4.3 Architecture of packages.

This folder contains custom packages and other packages found on the web that might be useful and are not taken in charge by Overleaf (see § 5.23). Currently, it holds:

- `local_basics.sty`, which contains useful macros for any document;
- `local_conversions.sty`, which contains conversion macros using the `fp` package for floating point arithmetic (see § 5.17);
- `local_kinematics.sty`, which contains useful kinematics / dynamics commands created for courses at the Université de Sherbrooke, such as ING225, GRO303, GRO640, and others;
- `local_physics.sty`, which contains useful commands for physics equations and mathematical proofs.

4.3 DATA/

This folder contains the images for figures, the code for listings, and the data for function plots in the document. For simplicity, I would recommend using the same architecture inside `data/` as the one in `src/`, as seen in figure 4.4:

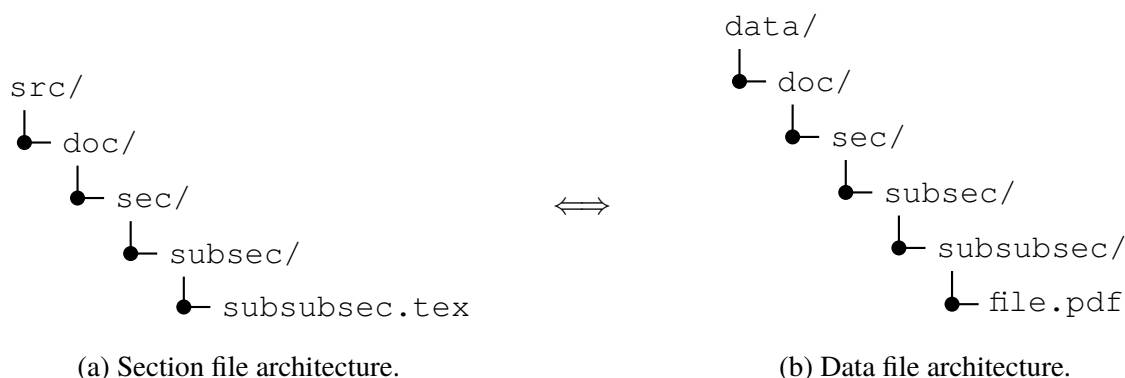


Figure 4.4 Comparison of section file and data file architectures.

It will create a hierarchy that may be bloated for small projects, but allows the project to change in scope and size pretty easily. Including the image in a figure becomes as simple as knowing the document name, looking at the document structure in the table of contents or the tree browser of the Overleaf project, and knowing the image file name (more details about figures in general in § 2.8):

```
\includegraphics[width=\figsize\columnwidth]{%
    {data/doc/sec/subsec/subsubsec/file.pdf}}
```

4.4 DEFS/

This folder contains \TeX files that defines new macros. Usually, they're used to create shortcuts for the document, such as `\title` for the document title on the cover page. One can also create custom environment or acronyms for words that are often used, although I would recommend the `glossaries` package if your words are technical and you would like to provide a list of acronyms and/or a glossary (see § 2.5.4).

4.5 SRC/

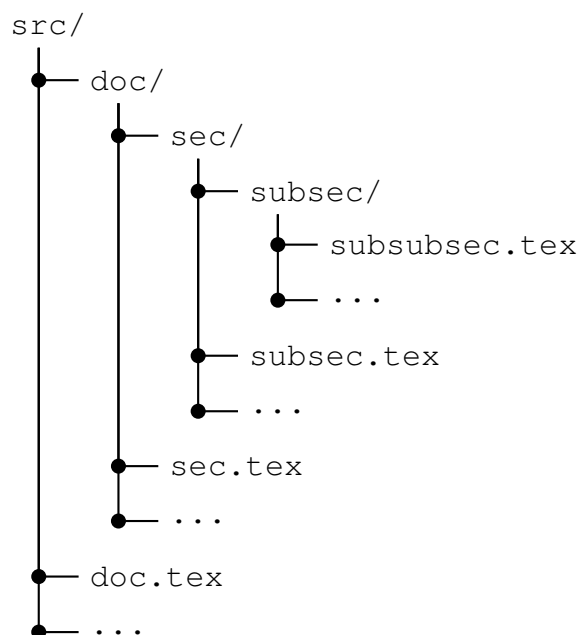


Figure 4.5 Architecture of section files.

This folder contains the \TeX files which represent the sections, subsections, and sub-subsections of a document. Every set of sections / subsections / sub-subsections is stored inside a folder that represents a single compiled document (e.g., `doc/`), which is defined inside a \TeX file at the top level of `src/` (e.g., `doc.tex`). Every section can have its own content defined in a \TeX file, and then every subsection inside that section does the same in a folder with the section name. Same principle applies for the sub-subsections. Top level files represent the compilation file of each document, which defines everything: settings, cover page, content, etc. In a previous version of my template, these top level files were stored in a separate folder on root, named `comp/`: this was quickly abandoned because I don't find any significant advantage in separating them from the rest of the document structure. As I said in § 4.3, it will create a hierarchy that seems bloated for small projects, but allows the project to change in scope and size pretty easily.

I suggest finding short handles for your section names, so it's easier to read in the tree browser on Overleaf. If your document has a lot of sections, or a section has a lot of subsections, etc., you can add a numbering scheme in your \TeX file names, though now your naming scheme is tied to the document structure and should be modified when the structure is altered.

In this template, we stop at three division levels, but it's possible to adapt it to include more

or less levels (more on that in § 2.4 and § 5.9).

4.5.1 TEMPLATES/

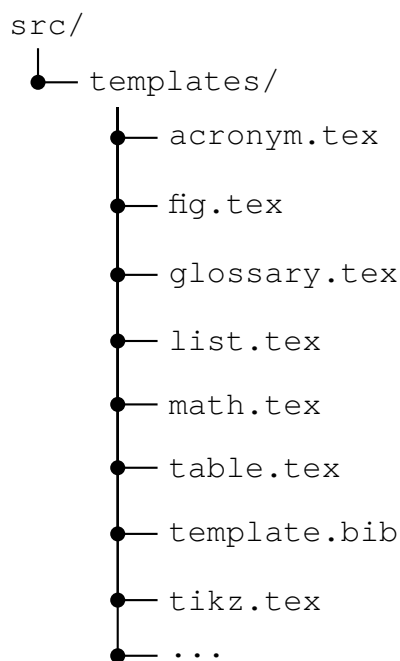


Figure 4.6 Architecture of template files.

This folder contains all the various templates for \LaTeX macros and environments that are not obvious to set up. By using templates every time we need to add a figure, table, equation, etc., we make sure that we respect the formatting rules we have to abide to (the caption positioning, for instance). You can also see the result of some of the templates in § 6. Acronyms, glossary entries, and sources are more akin to databases: they don't show any compilation result inside the document. Thus, the templates for these are more of a convention than a necessity.

4.6 TOOLS/

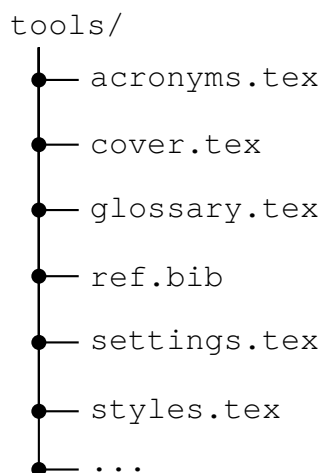


Figure 4.7 Architecture of tool files.

This folder contains settings, databases, and other global properties for the project. Current files are:

- `acronyms.tex` and `glossary.tex`, which hold the acronym and glossary entries. If the number of entries becomes overwhelming, it could be broken into small files nested in the `src/` architecture of the document they're used in;
- `ref.bib`, which holds the source data. If the number of sources becomes overwhelming, it could be broken into small files nested in the `src/` architecture of the document they're used in;
- `cover.tex`, which represent the global cover page. Different cover pages can be defined either in the compilation file of a document or in its `src/` hierarchy;
- `styles.tex`, which defines the various page styles that can be used in your documents (more on that in § 5.8.1, § 5.15.1, and § 5.22.1);
- `settings.tex`, which represents the global document preamble. It must be included in every document if you want it to compile, **no exceptions** (see § 5).

5 PREAMBLE

As you read along the past sections, you might have thought “this guide seems pretty well made, L^AT_EX must be magical!” If you *really* read along, you might have thought instead “this guide seems pretty well made... Where’s the real work hidden?” If you didn’t think of any sentence resembling these two, that means I need to work on my prediction skills. Still, to answer the second one: this is where the real work is hidden. The preamble, which encompasses everything that comes before the start of the document, allows us to define the style, the packages, the commands we will use later in writing the document. It can be created in a modular fashion with many different files that play together for given combinations, such as class, style, definition, and section files (see § 1.5). I have to admit that my preamble setup is not as optimized as I would like it to be since:

- I’m keeping everything in a single file because I’m constantly tweaking it, and managing feature changes in multiple files would get tiresome;
- I haven’t had the chance to work on document classes other than articles, thus my focus has not been on breaking down class and style options;
- bugs can appear depending on the load order of packages, which is really annoying and hurts flexibility.

Consequently, almost everything is contained within the `settings.tex` file, except for some custom commands and the `glossaries` package which doesn’t play nice when not called within the main document file. The following sections will detail the various considerations that went into making this guide, as well as more obscure macros that can help you understand better the inner workings.

5.1 DOCUMENT CLASS

```
11 \documentclass[letterpaper,12pt]{article}
```

Listing 5.1 Document class settings.

The `\documentclass{}` macro is *the essential L^AT_EX macro*. It is used to define a multitude of arguments for your documents, the necessary one being the document class in between braces. A comprehensive list of document classes openly available can be found on the CTAN [23]. Some standard ones include:

- `article`: articles in scientific journals, presentations, short reports, program documentation, invitations, etc. It's the document type most people default to;
- `beamer`: presentations, akin to PowerPoint for instance;
- `book`: longer books;
- `letter`: letters;
- `memoir`: based on the `book` class, but you can create any kind of document with it;
- `minimal`: as small as it can get. It only sets a page size and a base font. Mainly used for debugging purposes;
- `report`: longer reports with several chapters, small books, thesis, and more.

Some optional parameters—in between brackets—for standard classes include:

- document dimensions (`letterpaper`, `a4paper`, `legalpaper`, `executivepaper`, `a5paper`, `b5paper`). Defaults to `letter` or `A4` depending on the distribution and engine;
- font size (see § 2.3.2);
- number of sides (`oneside`, `twoside`);
- number of columns (`onecolumn`, `twocolumn`). Package `multicol` can extend that functionality to more than two columns;
- presence of a title page (`titlepage`, `notitlepage`);
- equation options (`fleqn` for left alignment of equations, `leqno` for equation labels to the left);
- chapter opening page (`openright`, `openany`);
- change print mode with `landscape`;
- change compiling option with `draft`;
- show the page frame with `showframe`. It comes in pretty handy to spot alignment and margin issues.

5.2 FOOTNOTES

```
17 \usepackage[multiple,bottom,flushmargin,hang]{footmisc}
```

Listing 5.2 Footnotes settings.

The `footmisc` package allows us to modify how footnotes are displayed. Some of the parameters can be tweaked within the language setup (see § 5.3.1), but most can be found here. The arguments selected in this document to standardize the formatting for the Université de Sherbrooke are:

- `multiple`: adds a separator when adding multiple footnotes at the same spot inside the text;
- `bottom`: forces footnotes at the bottom of the page;
- `flushmargin`: sets the footnote marker flush with, but just inside the margin from, the footnote content;
- `hang`: sets the footnote marker flush with the margin.

Loading the package before `babel` (see § 5.3) suppresses the behavior of the `multiple` argument. Loading it after suppresses the hyperlink functionality of footnotes provided by the `hyperref` package and adds a warning about the footnote mark redefinition. No known fix exists, so choose which functionality matters more to you... *or be the one to fix it!*

5.3 LANGUAGE

```

23 \usepackage[american,main=french]{babel}
24
25 %%% Recommended for use with "babel" and "polyglossia"
26 %%% to quote according to main language.
27 \usepackage{csquotes}

```

Listing 5.3 Language settings.

The `babel` package is generally recommended for monolingual documents since it is better maintained than alternatives and plays nicer with `biblatex` and `csquotes` (see § 5.4). The `polyglossia` package can be used for multilingual documents, although I have no experience with it. The command `\selectlanguage{}` has allowed me to switch between English and French without major issues.

With `babel`, you can select as many languages as you wish to support, with the `main` parameter being the one that modifies L^AT_EX presets and macros extensively. It calls a language “driver” for each of them. All “drivers” come with a different set of rules and parameters.

Then, we use the `csquotes` package to provide inline and display environments for quotes that follow the rules of the chosen main language.

5.3.1 LANGUAGE OPTIONS

```

628 \frenchsetup{
629     FrenchFootnotes    = false,
630     AutoSpaceFootnotes = false
631 }
632
633 \addto\captionsfrench{\def\figurename{Figure}}

```

```

634 \addto\captionsfrench{\def\tablename{Tableau}}
635 \addto\captionsfrench{\def\lstlistingname{Programme}}
636 % \addto\captionsfrench{\renewcommand{\abstractname}{Résumé}}
637 \addto\captionsfrench{\renewcommand{\refname}{Bibliographie}}
638 \addto\captionsfrench{\renewcommand{\contentsname}{Table des matières}}
639 \addto\captionsfrench{\renewcommand{\listfigurename}{Liste des figures}}
640 \addto\captionsfrench{\renewcommand{\listtablename}{Liste des tableaux}}
641 \addto\captionsfrench{\renewcommand{\lstlistlistingname}{Liste des programmes}}
642
643 \addto\captionsamerican{\def\lstlistingname{Listing}}
644 \addto\captionsamerican{\renewcommand{\listfigurename}{List of figures}}
645 \addto\captionsamerican{\renewcommand{\listtablename}{List of tables}}
646 \addto\captionsamerican{\renewcommand{\lstlistlistingname}{List of listings}}

```

Listing 5.4 Language options.

The `\frenchsetup` macro helps customize the formatting for French. The useful parameters in this case are:

- `FrenchFootnotes`: defines the french footnote style (indentation before the marker, non-superscript marker, point between the marker and the footnote content);
- `AutoSpaceFootnotes`: adds a non-breaking space between the last character and the footnote marker inside the text.

The other commands are there to define—or, in some cases, redefine—caption and section names for figures, tables, and listings, as well as renaming the bibliography section for any or all supported languages. You can do the same with any language of your choosing.

5.4 BIBLIOGRAPHY

```

33 \usepackage[
34     % backend    = biber,
35     % style      = authoryear-comp,
36     citestyle = numeric-comp,
37     sorting     = none,
38     alldates    = ymd
39 ]{biblatex}
40
41 %%% Add bibliography databases
42 \addbibresource{tools/ref.bib}
43 \addbibresource{src/templates/template.bib}
44
45 %%% Allow URL linebreaks in bibliography
46 \setcounter{biburllcpenalty}{1}
47 \setcounter{biburlucpenalty}{1}

```

Listing 5.5 Bibliography settings.

The `biblatex` package allows us to separate data from style by managing sources inside a database file. When invoking the package, several parameters can be used: a comprehensive list can be found in the `biblatex` documentation. [11] The most interesting ones for our application are:

- `backend`: selects `biber` by default. It's a more powerful backend, you should only choose `BibTeX` to support legacy databases;
- `style`: selects the general bibliography style;
- `citestyle`: selects the citation style. Value `numeric-comp` uses numbers to indicate citations;
- `sorting`: selects the sorting method. Value `none` sorts them by citation order inside the text;
- `alldates`: selects the display method for all date fields. Value `ymd` sets them according to ISO 8601 [24].

We then add the databases and modify counters for lower case and upper case bibliographical URL penalties such that then can be broken at any character. This avoids the issue of having a URL sticking out in the margin.

5.4.1 BIBLIOGRAPHY OPTIONS

```

49 %%% Replace parentheses with context-aware brackets/parentheses
50 \makeatletter
51   \newrobustcmd*{\parentexttrack}[1]{%
52     \begingroup
53       \blx@blxinit
54       \blx@setsfcodes
55       \blx@bibopenparen#1\blx@bibcloseparen
56     \endgroup
57   }
58   \AtEveryCite{%
59     \let\parentext      = \parentexttrack%
60     \let\bibopenparen   = \bibopenbracket%
61     \let\bibcloseparen = \bibclosebracket
62   }
63 \makeatother
64
65 %%% Modify citation to add hyperref to the author field
66 % \DeclareCiteCommand{\cite}
67 %   {\usebibmacro{prenote}}
68 %   {\usebibmacro{citeindex}%
69 %     \printtext[bibhyperref]{\usebibmacro{cite}}}
70 %   {}
71 %   {\multicitedelim}

```

```

72 %      {\usebibmacro{postnote}}
73 % \DeclareCiteCommand*{\cite}
74 %      {\usebibmacro{prenote}}
75 %      {\usebibmacro{citeindex}}%
76 %      \printtext[bibhyperref]{\usebibmacro{citeyear}}
77 %      }
78 %      {\multicitedelim}
79 %      {\usebibmacro{postnote}}
80 % \DeclareCiteCommand{\parencite}[\mkbibparens]
81 %      {\usebibmacro{prenote}}
82 %      {\usebibmacro{citeindex}}%
83 %      \printtext[bibhyperref]{\usebibmacro{cite}}
84 %      }
85 %      {\multicitedelim}
86 %      {\usebibmacro{postnote}}
87 % \DeclareCiteCommand*{\parencite}[\mkbibparens]
88 %      {\usebibmacro{prenote}}
89 %      {\usebibmacro{citeindex}}%
90 %      \printtext[bibhyperref]{\usebibmacro{citeyear}}
91 %      }
92 %      {\multicitedelim}
93 %      {\usebibmacro{postnote}}
94 % \DeclareCiteCommand{\footcite}[\mkbibfootnote]
95 %      {\usebibmacro{prenote}}
96 %      {\usebibmacro{citeindex}}%
97 %      \printtext[bibhyperref]{\usebibmacro{cite}}
98 %      }
99 %      {\multicitedelim}
100 %      {\usebibmacro{postnote}}
101 % \DeclareCiteCommand{\footcitetext}[\mkbibfootnotetext]
102 %      {\usebibmacro{prenote}}
103 %      {\usebibmacro{citeindex}}%
104 %      \printtext[bibhyperref]{\usebibmacro{cite}}
105 %      }
106 %      {\multicitedelim}
107 %      {\usebibmacro{postnote}}
108 % \DeclareCiteCommand{\textcite}
109 %      {\boolfalse{cbx:parens}}
110 %      {\usebibmacro{citeindex}}%
111 %      \printtext[bibhyperref]{\usebibmacro{textcite}}
112 %      }
113 %      {\ifbool{cbx:parens}
114 %      {\bibcloseparen\global\boolfalse{cbx:parens}}
115 %      {}%
116 %      \multicitedelim
117 %      }
118 %      {\usebibmacro{textcite:postnote}}
119
120 %%% Add comma between author and year
121 % \renewcommand*{\nameyeardelim}{\addcomma\space}

```

Listing 5.6 Bibliography options.

The first set of macros, replaces the “dumb” parentheses or brackets of citation commands with a context-aware version. This means that, when a citation is called inside parentheses, the citation itself will use brackets, and vice-versa. This helps avoid the chance of seeing double parentheses or double brackets.

The second set adds hyperlinking functionality to the author field inside a citation. When using the `authoryear-comp` citation style, the author name is not a clickable link to the source in the base `biblatex` package, only the year. These patches redefine the citation commands to include that author hyperlink.

The last two lines add a separator between the author name and the year in that same citation style, to comply with another suggested formatting by the Université de Sherbrooke.

5.5 DATE AND TIME

```
127 \usepackage{datetime2}
128 \DTMsetdatestyle{iso}
```

Listing 5.7 Date and time settings.

The `datetime2` package provides macros to format dates, times, and time zones. The command on the second line makes sure we follow the ISO 8601 norm for display of date and time inside the text (same constraint as in § 5.4).

5.6 FONTS

```
134 % \usepackage[latin1]{inputenc} % Older Windows machines
135 % \usepackage[utf8]{inputenc}   % Most 64-bit processors
136 % \usepackage{ae}               % T1 font emulation using CM fonts
137 \usepackage[T1]{fontenc}        % Font encoding
138 \usepackage{fontspec}           % XeLaTeX or LuaLaTeX
139
140 %%% Change font
141 \setromanfont{TeX Gyre Termes}
142 \setsansfont{TeX Gyre Adventor}
143 \setmonofont{TeX Gyre Cursor}
144
145 %%% Allow bold + small caps for Computer Modern
146 % \setmainfont{cmun}[
147 %   Extension      = .otf,
148 %   UprightFont    = *rm,
149 %   ItalicFont     = *ti,
150 %   BoldFont       = *bx,
151 %   BoldItalicFont = *bi,
152 % ]
```

```

153
154 %% Add symbols
155 % \usepackage{CJKutf8} % Chinese characters (loads inputenc.sty)
156 \usepackage{latexsym} % Support for LaTeX symbol fonts
157 \usepackage{textcomp} % Support for Text Companion fonts
158 \usepackage{hologo} % Collection of TeX-related logos
159 \usepackage{pifont} % Support for PostScript symbol fonts
160 \newcommand{\cmark}{\text{\ding{51}}} % Checkmark
161 \newcommand{\xmark}{\text{\ding{55}}} % X-Mark

```

Listing 5.8 Font settings.

The `fontenc` and `fontspec` packages are required to define the encoding and an interface for Apple advanced typography (AAT) and OpenType fonts on $\text{X}\text{\TeX}$ or $\text{Lua}\text{\TeX}$ (see § 1.1). The T1 encoding is 8-bit, comparatively to Computer Modern’s 7-bit encoding, OT1. T1 allows us to:

- hyphenate words with accented characters;
- copy and paste text output without any issue;
- avoid unexpected results with some special characters.

Other packages, such as `inputenc` and `ae` can be used if you need to support legacy $\text{L}\text{\TeX}$ code or run $\text{T}\text{\TeX}$ on older machines.

The various font setting commands are used to change the font: the standard one is Computer Modern, and the one in this document is a Times New Roman clone, per the requirement of the Université de Sherbrooke.

The next set of lines redefines the Computer Modern font to make the bold and small caps operations orthogonal, meaning they can be combined. That is not the case by default. These have been commented out since the document uses another font.

The last few lines add symbol packages for better display of particular symbols. The last two define new commands based on symbols available in the `pifont` package: the checkmark (✓), and the X mark (✗).

5.7 GEOMETRY

```

167 \usepackage{geometry}
168 \geometry{
169     left   = 1in,
170     right  = 1in,
171     top    = 1in,

```

```

172         bottom = lin
173     }
174
175     \setlength{\headheight}{15pt}
176     \setlength{\marginparsep}{6pt}
177     \setlength{\marginparpush}{\marginparsep}
178     \setlength{\marginparwidth}{\dimexpr lin-\marginparsep-\marginparpush}
179
180     %%% Change page layout in the middle of a document
181     % \usepackage{changepage}
182
183     %%% Show summary of document layout
184     \usepackage{layout}
185
186     %%% Rotation
187     % \usepackage{lscape} % Rotate content
188     % \usepackage{pdflscape} % Rotate page
189
190     %%% Change geometry with landscape parameter
191     \makeatletter
192     \def\ifGm@preamble#1{\@firstofone}
193     \appto\restoregeometry{%
194         \pdfpagewidth=\paperwidth
195         \pdfpageheight=\paperheight}
196     \apptocmd\newgeometry{%
197         \pdfpagewidth=\paperwidth
198         \pdfpageheight=\paperheight}{}{}
199     \apptocmd\loadgeometry{%
200         \pdfpagewidth=\paperwidth
201         \pdfpageheight=\paperheight}{}{}
202     \makeatother

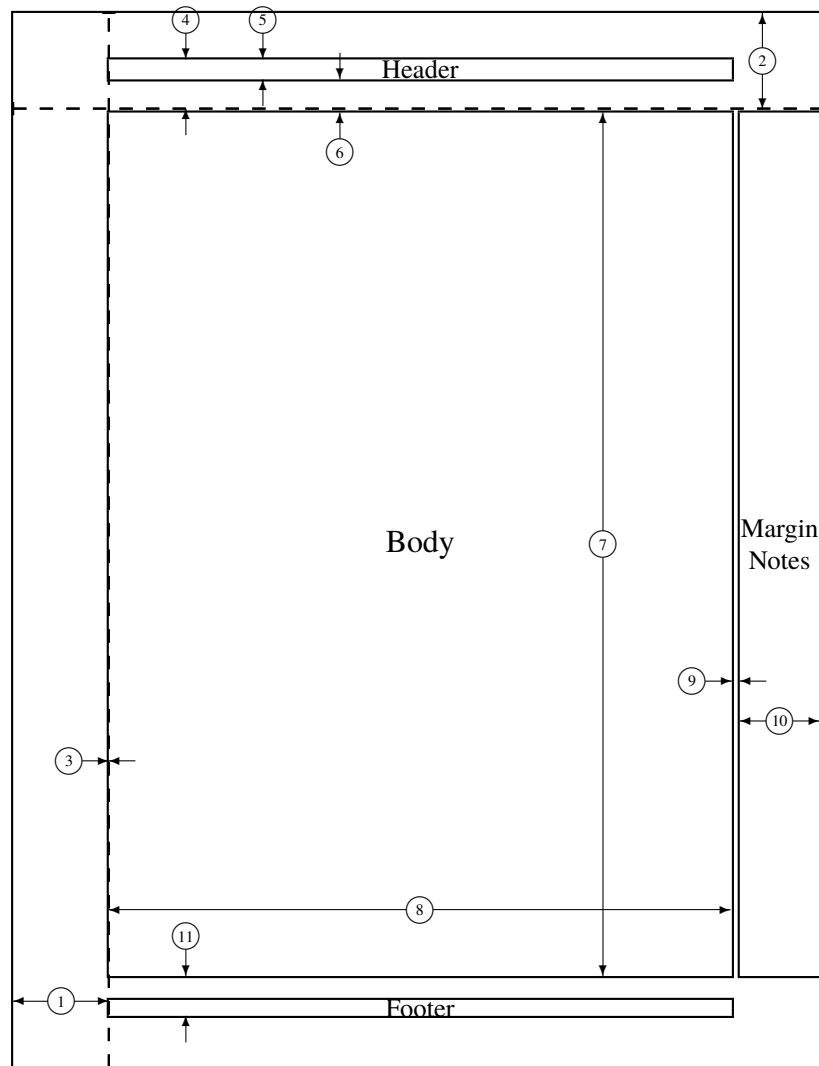
```

Listing 5.9 Geometry settings.

The `geometry` package defines the size and margins of our document. We can then set the length of other dimensions, such as the margin separation, the margin push, etc. A visual representation of those changes can be seen in figure 5.1 generated with the `layout` package.

Packages such as `changepage`, `lscape`, and `pdflscape` are all superseded by the macro defined in the last few lines. It overloads `geometry` commands to add the ability to define new layouts that can be loaded on individual pages or sets of pages:

- create layout with `\newgeometry{}`;
- save layout with `\savegeometry{}`;
- restore original layout with `\restoregeometry`;
- load new layout with `\loadgeometry{}`.



1 one inch + <code>\hoffset</code>	2 one inch + <code>\voffset</code>
3 <code>\oddsidemargin = 0pt</code>	4 <code>\topmargin = -37pt</code>
5 <code>\headheight = 15pt</code>	6 <code>\headsep = 25pt</code>
7 <code>\textheight = 650pt</code>	8 <code>\textwidth = 469pt</code>
9 <code>\marginparsep = 6pt</code>	10 <code>\marginparwidth = 60pt</code>
11 <code>\footskip = 30pt</code>	<code>\marginparpush = 6pt</code> (not shown)
<code>\hoffset = 0pt</code>	<code>\voffset = 0pt</code>
<code>\paperwidth = 614pt</code>	<code>\paperheight = 794pt</code>

Figure 5.1 Document layout information.

5.8 TYPESETTING

```
208 \usepackage{listings}
```

Listing 5.10 Listing settings.

The `listings` package provides utilities to typeset programs within \LaTeX . We can define programming languages or extend them (see § 5.8.1), import code files, and label typeset chunks of code just like we would with figures, tables, and equations.

5.8.1 TYPESETTING STYLE

```
32 \lstset{
33     % backgroundcolor = \color[HTML]{FFFFFF},
34     basicstyle       = \scriptsize\bfseries\ttfamily\color{AtomOneLightBlack},
35     breakatwhitespace = false,
36     breaklines       = true,
37     captionpos       = b,
38     % columns         = fullflexible,
39     commentstyle      = \color{AtomOneLightOrange},
40     % deletekeywords  = {},
41     % escapeinside    = {\%*}{*} },
42     % extendedchars   = true,
43     % fillcolor       = \color{white},
44     % firstnumber     = 1000,
45     % frame           = none,
46     % framerule       = 0.4pt,
47     % framesep        = 4.5mm,
48     % framexleftmargin = 2.5mm,
49     % identifierstyle = \idstyle,
50     % keepspaces      = true,
51     keywordstyle      = \color{AtomOneLightPurple},
52     % language        = TeX,
53     numbers           = left,
54     numbersep        = 10pt,
55     numberstyle       = \scriptsize\ttfamily\color{AtomOneLightGray},
56     % rulecolor       = \color{AtomOneLightBlack},
57     sensitive        = true,
58     showspaces       = false,
59     showstringspaces = false,
60     showtabs         = false,
61     stepnumber       = 1,
62     stringstyle      = \color{AtomOneLightGreen},
63     tabsize          = 4,
64     % title           = \lstname
65 }
66
67 \lstdefinelanguage{LaTeX}{%
68     language      = {[LaTeX]TeX},
69     moretexcs     = {
```

```

70 abstractname, aboverulesep, acronymtype, addbibresource, addto, addtocontents,
71 appto, apptocmd, arraybackslash, AtBeginDocument, AtEveryCite, belowrulesep,
72 bibclosebracket, bibcloseparen, bibopenbracket, bibopenparen, bottomrule,
73 captionsamerican, captionsfrench, CB, CC, cftbeforesecskip, cftbeforesubsecskip,
74 cftbeforesubsubsecskip, cftbeforetoctitleskip, cftdot, cftdotfill, cftdotsep,
75 cftfigindent, cftfignumwidth, cftlistingsindent, cftlistingsnumwidth,
76 cftloftitlefont, cftloltitlefont, cftlottitlefont, cftsecfont, cftsecindent,
77 cftsecleader, cftsubsecfont, cftsubsecindent, cftsubsubsecfont,
78 cftsubsubsecindent, cfttabindent, cfttoctitlefont, cmark, color, columnA,
79 columnB, columnC, columnD, columnE, columnF, columnG, columnH, columnI, columnJ,
80 columnK, columnL, columnM, columnN, columnO, columnP, contentsname,
81 convertexactly, convertwithrounding, counterwithin, CoverName, CT, definecolor,
82 deriv, dimexpr, ding, doublevbar, DTMsetdatestyle, end, endfirsthead, endhead,
83 fancyfoot, fancyhead, fancyheadoffset, fancyhf, fancypagestyle, figsize,
84 figurename, footrulewidth, frenchsetup, geometry, glossaryheader, glossentry,
85 Glossentrydesc, glossentryname, gls, glsentrydesc, glsentryitem, glsentrytext,
86 glsgroupskip, glspagelistwidth, glsseelist, glssubentryitem, glstarget,
87 glstextformat, headrulewidth, hypersetup, includegraphics, inp, labelitemi, LB,
88 LC, listfigurename, listtablename, loadgeometry, loadglsentries,
89 lstdefinlanguage, lstlistingname, lstlistlistingname, lstset, LT,
90 makeglossaries, mathcal, multirow, mydepth, myheight, mylength, mywidth,
91 newbullet, newcolumntype, newgeometry, newglossaryentry, newglossarystyle,
92 newlistof, newrobustcmd, numexpr, onebp, onecc, onecm, onedd, oneem, oneex,
93 onein, onemm, onepc, onept, onesp, paragraph, parenttext, parenttexttrack,
94 pdfpageheight, pdfpagewidth, pgfplotsset, RB, RC, refname, restoregeometry, RT,
95 setlength, setlist, setmainfont, setmonofont, setromanfont, setsansfont, sfrac,
96 subglossentry, subsection, subsubsection, tablename, tcbset, text, textcolor,
97 theequation, thefigure, thelstlisting, theparagraph, thesection, thesubfigure,
98 thesubsection, thesubsubsection, thetable, tikzset, titleformat, titlespacing,
99 usetikzlibrary, vbar, widthof, xintdiv, xintIrr, xintMul, xintReverseOrder,
100 xintRound, xmark
101 },
102 % emph = [1]{
103 % bottom, empty, figure, landscape, left, longtable,
104 % lstlisting, plain, right, subfigure, top
105 % },
106 % emphstyle = [1]{\color{AtomOneLightYellow},
107 alsoletter = {\&},
108 emph = [2]{\&},
109 emphstyle = [2]{\color{AtomOneLightBlue}},
110 }

```

Listing 5.11 Listing style.

The `\lstset` macro defines the global typesetting parameters of the document. The `\lstdefinlanguage` command can be used to create our own programming language or, in this case, extend the definition of \LaTeX to allow syntax highlighting on more keywords. For more

information on the package and how to use it, feel free to consult the CTAN documentation [25].

5.9 TABLE OF CONTENTS AND LISTS

```

214 \usepackage[titles]{tocloft}
215
216 %%% Add list of listings to tocloft
217 \makeatletter
218   \begingroup
219     \let\newcounter\@gobble\let\setcounter\@gobbletwo
220     \globaldefs\@ne \let\c@l@depth\@ne
221     \newlistof{listings}{lol}{\lstlistlistingname}
222   \endgroup
223   \let\l@lstlisting\l@listings
224   % \AtBeginDocument{\addtocontents{lol}{\protect}}
225 \makeatother
226
227 %%% Change title style
228 % \renewcommand{\cfttoctitlefont}{\normalfont\Large\bfseries\scshape}
229 % \renewcommand{\cftloftitlefont}{\normalfont\Large\bfseries\scshape}
230 % \renewcommand{\cftlottitlefont}{\normalfont\Large\bfseries\scshape}
231 % \renewcommand{\cftloltitlefont}{\normalfont\Large\bfseries\scshape}
232
233 % \setlength{\cftbeforetoctitleskip}{6pt}
234
235 %%% Change number column width
236 \renewcommand{\cftlistingsnumwidth}{\cftfignumwidth}
237
238 \renewcommand{\cftsecfont}{\bfseries} % Change section display
239 % \renewcommand{\cftsubsecfont}{\scshape} % Change subsection display
240 % \renewcommand{\cftsubsubsecfont}{\scshape} % Change subsubsection display
241
242 \renewcommand{\cftsecleader}{\cftdotfill{\cftdotsep}} % Add dots
243 % \renewcommand{\cftdot}{} % Remove dots
244
245 % \setlength{\cftsecindent}{0em}
246 % \setlength{\cftsubsecindent}{1.5em}
247 % \setlength{\cftsubsubsecindent}{3.75em}
248 \setlength{\cftfigindent}{0pt}
249 \setlength{\cfttabindent}{0pt}
250 \setlength{\cftlistingsindent}{0pt}
251
252 \setlength{\cftbeforesecskip}{6pt}
253 \setlength{\cftbeforesubsecskip}{0pt}
254 \setlength{\cftbeforesubsubsecskip}{0pt}
255
256 \setcounter{tocdepth}{2}
257 \setcounter{secnumdepth}{3}

```

Listing 5.12 Table of contents and lists settings.

The `tocloft` package allows us modify the appearance of the ToC and the plethora of lists we could create: LoF, LoL, LoT, and more. The `titles` argument causes the titles to be typeset with the default \LaTeX methods. This means that all titles follow global setup and we don't need to change them manually.

Lines 216–225 add the LoL to `tocloft`. The package `listings` creates the LoL automatically, but we need the command to allow modification through `tocloft`'s implementation. This is a good time to mention that the LoL title will not be positioned correctly if the package `listings` is loaded after the package `float`: there will be extra vertical space that we can't get rid of.

Line 236 makes sure that the allocated space for numbered bookmarks on the LoL is the same as other lists.

Lines 238–240 modify the font formatting for the various section levels of the document. In this case, we only need to tweak the series of the section (see § 2.3.1) per the requirement of the Université de Sherbrooke.

Lines 242 and 243 control the dotted leader lines inside the ToC and the lists. This is a matter of preference: feel free to use either option to your heart's content. If you use dotted header lines, the reason to avoid the dot at the end of a caption for the various lists (see § 2.8, § 2.9.3, and § 2.10) becomes evident: if we leave a dot in the caption, the spacing is non-uniform with the header lines.

Lines 245–249 control the indentation of the section levels and floats. Subsection are indented such that the numbered bookmark aligns with the section text, ditto for the sub-subsection bookmark with the subsection text. Float references are left without indentation.

Lines 252–254 control the vertical space around the section levels in the ToC. In our case, we only need extra spacing around sections to follow the guidelines of the Université de Sherbrooke.

Line 256 defines the number of levels that appear in the ToC, and line 257 defines the number of section levels that receive numbered bookmarks.

For more configuration macros with `tocloft`, refer to its CTAN documentation [26].

5.10 FIGURES

```
263  %% Optional arguments for "\includegraphics".
264  %% Extension of "graphics".
265  \usepackage{graphicx}
266
```

```

267 %%% Captions
268 \usepackage[labelsep=quad]{caption} % Floating environments
269 \usepackage[labelformat=simple]{subcaption} % Subfigures and subtables
270
271 \usepackage{float} % Improved interface for floating objects
272 \usepackage{morefloats} % Increased number of floats
273 % \usepackage{wrapfig} % Wrap figures in text
274
275 %%% Page fraction for float environments vs text
276 \renewcommand{\topfraction}{0.95}
277 \renewcommand{\bottomfraction}{0.95}
278 \renewcommand{\textfraction}{0.05}
279 % \renewcommand\floatpagefraction{0.8}
280
281 %%% Global figure size
282 \newcommand{\figsize}{0.85}

```

Listing 5.13 Figure settings.

The `graphix` package is used to extend the functionality of the base `graphics` package, by providing a key-value interface for optional arguments of the `\includegraphics[]{}{}` command for instance.

The `caption` package is used to customize the captions in floating environments, such as a figure, a table, or a listing. The `labelsep` parameter sets the spacing between the numbered bookmark and the caption text. The `subcaption` package extend that functionality to subfigures, subtables, etc. We make sure the subcaption is displayed the same way as the caption by specifying the `labelformat` parameter.

The `float` package gives us improved macros for floating objects, such as the `H` float specifier for figures (see table 2.19). The `morefloats` package increase the number of floating objects the engine can handle, which comes in handy for large documents.

The `wrapfig` package is currently commented out, but anyone that wants to wrap figures in text can include it in the preamble and use its custom environment.

Lines 275–279 change the page fraction allowed for floating objects. The smaller the portion of the page, the more inclined a figure will be to move around and jump to surrounding pages. This only applies to the standard float specifiers (see table 2.19).

The `\figsize` command sets a ratio between 0 and 1 to modulate the figure width globally (see § 2.8).

5.11 COLORS

```

288 \usepackage[table,dvipsnames,svgnames,xllnames]{xcolor}
289 \usepackage[most]{tcolorbox} % Colored boxes
290
291 \tcbset{%
292     on line,
293     highlight math style = {enhanced},
294     boxsep    = 4pt,
295     left      = 0pt,
296     right     = 0pt,
297     top       = 0pt,
298     bottom    = 0pt,
299     colframe  = white,
300     colback   = gray!50
301 }
302
303 %%% Atom One Light Theme
304 \definecolor{AtomOneLightBlack}{HTML}{383A42}
305 \definecolor{AtomOneLightGray}{HTML}{A0A1A7}
306 \definecolor{AtomOneLightYellow}{HTML}{C18401}
307 \definecolor{AtomOneLightOrange}{HTML}{E45649}
308 \definecolor{AtomOneLightRed}{HTML}{AB2525}
309 \definecolor{AtomOneLightPurple}{HTML}{A626A4}
310 \definecolor{AtomOneLightBlue}{HTML}{4078F2}
311 \definecolor{AtomOneLightGreen}{HTML}{50A14F}

```

Listing 5.14 Color settings.

The `xcolor` package provides driver-independent access to color tints, shades, tones, and mixes. The various arguments are there to support other packages and load predefined colors.

The `tcolorbox` package provides an environment for colored and framed text boxes. The global parameters are set within the `\tcbset` command.

Lines 303–311 define custom colors to create a theme that can be used to define other styles. For instance, these colors define the listing style (see § 5.8.1).

5.12 TIKZ

```

317 %%% Graphic elements
318 % \usepackage{tikz}
319 % \usepackage[RPvoltages]{circuitikz} % Loads TikZ
320 \usepackage{forest} % Loads TikZ
321
322 %%% Node positioning, arithmetic, and others
323 \usetikzlibrary{positioning}
324 \usetikzlibrary{calc}
325 % \usetikzlibrary{shapes}
326 % \usetikzlibrary{arrows}
327 % \usetikzlibrary{patterns}

```

```

328 % \usetikzlibrary{tikzmark}
329 \usetikzlibrary{decorations.pathreplacing}
330 \usetikzlibrary{matrix}
331 \usetikzlibrary{fit}
332 \usetikzlibrary{backgrounds}
333
334 %%% Scientific graphs
335 \usepackage{pgfplots}
336 \pgfplotsset{compat=1.17} % Version for TeX Live 2020
337
338 % \makeatletter
339 %     \newdimen\multi@col@width
340 %     \newdimen\multi@col@margin
341 %     \newcount\multi@col@count
342 %     \multi@col@width=0pt
343
344 %     \tikzset{
345 %         multicol/.code = {%
346 %             \global\multi@col@count=#1\relax
347 %             \global\let\orig@pgfmatrixendcode=\pgfmatrixendcode
348 %             \global\let\orig@pgfmatrixemptycode=\pgfmatrixemptycode
349 %             \def\pgfmatrixendcode##1{%
350 %                 \orig@pgfmatrixendcode##1%
351 %                 \pgfutil@tempdima=\pgf@picmaxx
352 %                 \global\multi@col@margin=\pgf@picminx
353 %                 \advance\pgfutil@tempdima by -\pgf@picminx
354 %                 \divide\pgfutil@tempdima by #1\relax
355 %                 \global\multi@col@width=\pgfutil@tempdima
356 %                 \pgf@picmaxx=.5\multi@col@width
357 %                 \pgf@picminx=-.5\multi@col@width
358 %                 \global\pgf@picmaxx=\pgf@picmaxx
359 %                 \global\pgf@picminx=\pgf@picminx
360 %                 \gdef\multi@adjust@position{%
361 %                     \setbox\pgf@matrix@cell=\hbox\bgroup
362 %                     \hfil\hskip-1.5\multi@col@margin
363 %                     \hfil\hskip-.5\multi@col@width
364 %                     \box\pgf@matrix@cell
365 %                     \egroup
366 %                 }%
367 %                 \gdef\multi@temp{\aftergroup\multi@adjust@position}%
368 %                 \aftergroup\multi@temp
369 %             }
370 %             \gdef\pgfmatrixemptycode{%
371 %                 \orig@pgfmatrixemptycode
372 %                 \global\advance\multi@col@count by -1\relax
373 %                 \global\pgf@picmaxx=.5\multi@col@width
374 %                 \global\pgf@picminx=-.5\multi@col@width
375 %                 \ifnum\multi@col@count=1\relax
376 %                     \global\let\pgfmatrixemptycode=\orig@pgfmatrixemptycode
377 %                 \fi
378 %             }
379 %         }

```

```

380 %      }
381 % \makeatother

```

Listing 5.15 TikZ settings.

The `tikz` package provides an interface to draw graphics directly in the document. It is currently loaded through the `forest` package, which is used in § 4 to draw architecture trees. In other reports, I had to draw electrical circuits: I used the `circuitikz` package instead. It all depends on your needs. For more information on the package, check out the CTAN documentation [21].

TikZ can be extended with libraries, which can be called separately as needed. They provide additional macros to facilitate the creation of complex graphics. Currently, the libraries `positioning`, `calc`, `decorations`, `matrix`, `fit`, and `backgrounds` are used for architecture trees and templates (see § 6.5).

The `pgfplots` package helps create 2D/3D function plots, which is a really neat utility for lab reports for instance. It can be cumbersome to tweak the graphical output according to one’s formatting requirements, but the end result is beautiful. It can also read data from a file (e.g., `.dat`), an ability that provides easy modification if results change over time. The version is specified with the `\pgfplotsset{}` command. For more information, consult the CTAN documentation [27].

The remainder of the code (lines 338–381) defines a multi-column utility for TikZ tables.

5.13 TEXT

```

387 %% Affects line breaks
388 \tolerance = 1000 % Allows all stretched inter-word spaces
389 % \hyphenpenalty = 10000 % No hyphenation
390
391 %% Paragraph indent, skip, and line spacing
392 \setlength{\parindent}{1.25cm}
393 \setlength{\parskip}{6pt}
394 \linespread{1.3}

```

Listing 5.16 Text settings.

The `\tolerance` command sets the allowed stretching. \TeX assigns “badness” values to whatever operation it has to make in order to find the best spot to to break to the next line. The sum of the values for each line gives us a total “badness” that gets compared to the tolerance: the first result below the tolerance is the chosen line break. You can read more about this process in Donald E. Knuth’s book on \TeX [4].

The `\hyphenpenalty` command sets the “badness” level above which T_EX can look into hyphenation to reduce the line breaking error. Setting it to the maximum value of 10,000 avoids hyphenation entirely.

The `\parindent`, `\parskip`, and `\linespread` lengths respectively define the paragraph indentation, the extra vertical space in between paragraphs, and the line spacing.

5.14 CONDITIONALS

```
400 \usepackage{xifthen}
```

Listing 5.17 Conditionals settings.

The `xifthen` package extends the functionality of the `ifthen` package to allow for more complex tests of conditionals (variable declaration, equality of variables, find a void string, etc.). It is currently used in the cover page for reports, which checks if an author name is empty or not. If not, it is added to the cover page.

5.15 HEADER AND FOOTER

```
406 \usepackage{fancyhdr}
```

```
407 \usepackage{lastpage}
```

Listing 5.18 Header and footer settings.

The `fancyhdr` package provides macros to customize the appearance and easily swap between styles of headers and footers. You can select one with commands `\thispagestyle` for a single page or `\pagestyle` for the following pages. Standard styles include `empty`, `plain`, `headings`, and `myheadings`. You can also define custom styles (see § 5.15.1).

The `lastpage` package provides a command to refer to the total number of pages of your document. This is currently used in the footer of this document.

5.15.1 HEADER AND FOOTER STYLE

```
12 \renewcommand{\subsectionmark}[1]{%
13   \markright{\MakeUppercase{\thesubsection.\ #1}}
14 }
15
16 \fancypagestyle{udes}{
17   \fancyhf{}%
18   % \fancyhead[L]{\leftmark}%
19   % \fancyhead[R]{\rightmark}%
20   % \fancyhead[C]{\MakeUppercase{\title}}%
```

```

21 \fancyfoot[C]{\thepage{} of \pageref*{LastPage}}%
22 % \fancyfoot[R]{\includegraphics[scale=0.15]{img/udes/3it_footer.pdf}}%
23 \renewcommand{\headrulewidth}{0pt}%
24 % \renewcommand{\footrulewidth}{\arrayrulewidth}
25 \fancyheadoffset{0pt}
26 }

```

Listing 5.19 Header and footer style.

This is a custom header and footer style for my reports. We first start by resetting the default formatting with `\fancyhf{}`. We can then declare anything we want inside the header or the footer with `\fancyhead{}` or `\fancyfoot{}` respectively. The `L / C / R` option sets the position (left, center, right). The commands `\leftmark` and `\rightmark` refer to the top and bottom section levels respectively. The rulers `\headrulewidth` and `\footrulewidth` can be redefined or commented out if you want separators for your header and footer or not. Finally, the command `\fancyheadoffset{}` is reset to allow automatic adjustment of the header and footer when changing the geometry of the document (see § 5.7).

5.16 TABLES

```

413 \usepackage{multirow}      % Multi-row cells
414 \usepackage{multicol}     % Multi-column cells
415 \usepackage{longtable}    % Multi-page tables
416 \usepackage{booktabs}     % Table optimization
417 \usepackage{array}        % Extension of column formats
418 % \usepackage{nicematrix}  % Nicer tables with colored cells
419 % \usepackage{makecell}    % Column head and multilined cells
420 % \usepackage{supertabular} % Multi-page tables
421
422 %%% Booktabs rule separation
423 % \setlength{\aboverulesep}{0pt}
424 % \setlength{\belowrulesep}{0pt}
425
426 %%% Padding
427 \renewcommand{\arraystretch}{\baselinestretch}
428 \setlength{\tabcolsep}{0.01\linewidth}
429
430 %%% Spacing at the end of a table head
431 \makeatletter
432 \def\endhead{\bottomrule[\arrayrulewidth] \LT@end@hd@ft\LT@head}
433 \makeatother
434 \makeatletter
435 \def\endfirsthead{\bottomrule[\arrayrulewidth] \LT@end@hd@ft\LT@firsthead}
436 \makeatother
437
438 %%% Column types
439 \newcolumntype{\LC}[1]{%

```



```

440     >{\raggedright\let\newline\\\arraybackslash\hspace{0pt}}m{#1}
441 }
442 \newcolumnntype{\CC}[1]{%
443     >{\centering\let\newline\\\arraybackslash\hspace{0pt}}m{#1}
444 }
445 \newcolumnntype{\RC}[1]{%
446     >{\raggedleft\let\newline\\\arraybackslash\hspace{0pt}}m{#1}
447 }
448
449 \newcolumnntype{\LB}[1]{%
450     >{\raggedright\let\newline\\\arraybackslash\hspace{0pt}}b{#1}
451 }
452 \newcolumnntype{\CB}[1]{%
453     >{\centering\let\newline\\\arraybackslash\hspace{0pt}}b{#1}
454 }
455 \newcolumnntype{\RB}[1]{%
456     >{\raggedleft\let\newline\\\arraybackslash\hspace{0pt}}b{#1}
457 }
458
459 \newcolumnntype{\LT}[1]{%
460     >{\raggedright\let\newline\\\arraybackslash\hspace{0pt}}p{#1}
461 }
462 \newcolumnntype{\CT}[1]{%
463     >{\centering\let\newline\\\arraybackslash\hspace{0pt}}p{#1}
464 }
465 \newcolumnntype{\RT}[1]{%
466     >{\raggedleft\let\newline\\\arraybackslash\hspace{0pt}}p{#1}
467 }
468
469 %%% Custom lengths
470 \newlength{\vbar}
471 \setlength{\vbar}{\arrayrulewidth}
472 \newlength{\doublevbar}
473 \setlength{\doublevbar}{\dimexpr 2pt+2\arrayrulewidth}
474 \newlength{\columnA}
475 \newlength{\columnB}
476 \newlength{\columnC}
477 \newlength{\columnD}
478 \newlength{\columnE}
479 \newlength{\columnF}
480 \newlength{\columnG}
481 \newlength{\columnH}
482 \newlength{\columnI}
483 \newlength{\columnJ}
484 \newlength{\columnK}
485 \newlength{\columnL}
486 \newlength{\columnM}
487 \newlength{\columnN}
488 \newlength{\columnO}
489 \newlength{\columnP}

```

Listing 5.20 Table settings.

The `multirow` and `multicolumn` packages are obviously used to provide multi-row and multi-column functionalities for tables. The `longtable` package allows the creation of tables that support page breaks. The `booktabs` package provides quality enhancement and optimisation to tables, especially for those trying to avoid the “prison” style with many vertical and horizontal bars. The `array` package extends the available column types and gives more format specification options. Packages `nicematrix`, `makecell`, and `supertabular` are not used in this document at the moment. New packages could be introduced to supplement or replace the existing ones as I keep looking for the simplest solution that supports page breaks, multi-row and multi-column functionalities, text wrapping, and horizontal and vertical alignments.

Lines 422–424, which are currently commented out, define the separation above and below rulers for tables created with `booktabs` utilities.

Lines 426–428 define the stretching factor of table cells, which here is the same as the line spacing, and the padding on each side of a cell.

Lines 430–436 define custom spacing after table headers, which gives the appearance of a floating header. This is just a matter of personal preference.

Lines 438–467 defines 9 new column types to accommodate all of your alignment needs (see § 2.12.1):

- `\raggedleft / \centering / \raggedright`: defines the horizontal alignment;
- `b{ } / m{ } / p{ }`: selects the right column type from the `array` package to provide vertical alignment;
- `\let\newline\\arraybackslash\hspace{0pt}`: restore text wrapping and line breaking functionalities.

Lines 469–489 define custom lengths, such as single and double vertical bars as well as custom column widths for latter use in tables (see § 2.12.1).

5.17 MATHEMATICS AND ARITHMETIC

```

495 \usepackage{mathtools} % Extension of "amsmath"
496 \usepackage{amssymb}   % Math symbols
497 \usepackage{upgreek}   % Upright greek symbols
498 \usepackage{esint}     % Integral symbols for CM fonts
499 \usepackage{siunitx}    % Units in math mode
500 \usepackage{nicefrac}   % Fraction alternative
501 \usepackage{cancel}     % Cancellation symbols
502 % \usepackage{newtxmath} % Math symbols

```

```

503 % \usepackage{bm}           % Bold math symbols
504 % \usepackage{mdsymbol}     % Math symbols for Adobe Myriad Pro font
505 % \usepackage{MnSymbol}     % Math symbols for Adobe Minion Pro font
506 % \usepackage{icomma}       % Comma as decimal separator
507
508 \usepackage{xintfrac} % Operations on long numbers
509 % \usepackage{fp}         % Floating point arithmetic
510
511 %%% Define all typographic dimensions
512 \def\onecm {1}
513 \def\onemm {1/10}
514 \def\onein {2.54}
515 \def\onept {2.54/72.27}
516 \def\onebp {2.54/72}
517 \def\onepc {\xintMul {12}{\onept}}
518 \def\oneex {\xintMul {\the\numexpr\dimexpr 1ex\relax\relax}{\onesp}}
519 \def\oneem {\xintMul {\the\numexpr\dimexpr 1em\relax\relax}{\onesp}}
520 \def\onedd {\xintMul {1238/1157}{\onept}}
521 \def\onecc {\xintMul {12}{\onedd}}
522 \def\onesp {\xintMul {1/65536}{\onept}}
523
524 %%% Create conversion functions
525 \makeatletter
526 \def\convertexactly #1\to #2%
527   {\xintIrr{\convertexactly@ #1\to {#2}}}%
528 \def\convertwithrounding #1#2\to #3%
529   {\xintRound {#1}{\convertexactly@ #2\to {#3}}}%
530 \def\convertexactly@ #1\to
531 {%
532   \romannumeral0%
533   \expandafter\expandafter\expandafter
534   \convertexactly@a
535   \xintReverseOrder {#1}\Z
536 }%
537 \def\convertexactly@a #1%
538 {%
539   \ifcat\noexpand #1\relax
540     \expandafter \convertexactly@b
541   \else
542     \expandafter \convertexactly@c
543   \fi #1%
544 }%
545 \def\convertexactly@b #1#2\Z #3%
546 {%
547   \xintdiv {\xintMul {\xintReverseOrder{#2}}{\csname one#1\endcsname}}
548   {\csname one#3\endcsname}%
549 }%
550 \def\convertexactly@c #1#2#3\Z #4%
551 {%
552   \xintdiv {\xintMul {\xintReverseOrder{#3}}{\csname one#2#1\endcsname}}
553   {\csname one#4\endcsname}%
554 }%

```

Listing 5.21 Mathematics and arithmetic settings.

Since \TeX was created mostly for proper display of equations in documents, there is a plethora of packages to choose from, the ones in use at the moment being:

- `mathtools`: provides extensible symbols and more environments to supplement the \LaTeX -required `amsmath` package;
- `amssymb`: provides extra symbols and font utilities for mathematical typesetting;
- `upgreek`: adds upright Greek letters;
- `esint`: extends the set of integral symbols;
- `siunitx`: consistent interface to define the number and unit conventions;
- `nicefrac`: adds a new fraction style;
- `cancel`: add cancellation symbols for mathematical developments.

As for the arithmetic side of things, the only package in use is `xintfrac`. It provides operations on long numbers, which increases the precision of various calculations. Its commands are use in lines 511–555 to define units of length and conversion macros to create table 2.4. Its use would be very limited in other cases, but you could still want to create functions for quick and flexible arithmetic.

5.18 HYPERLINKS

```

561 % \usepackage{url}
562 \usepackage[pdfpagelabels=true]{hyperref}
563
564 \hypersetup{
565     % pdfa,
566     % pdftex,
567     bookmarksnumbered,
568     bookmarksopen      = true,
569     bookmarksopenlevel = \arabic{tocdepth},
570     bookmarksdepth     = \arabic{secnumdepth},
571     pdfhighlight       = /N,
572     colorlinks         = false,
573     urlcolor           = BrickRed,
574     linkcolor          = Blue,
575     citecolor          = BrickRed,
576     linktoc            = all
577 }
578
579 %%% Back reference

```

```

580 % \usepackage[hyperpageref]{backref}
581 % \renewcommand{\backref}[1]{[cf. p. #1]}
582
583 %%% More hyperref option
584 \usepackage{bookmark}
585
586 %%% Name of cover page
587 \newcommand{\CoverName}{Cover}

```

Listing 5.22 Hyperlink settings.

The `hyperref` package provides the hyperlink functionality used everywhere: URLs, glossary and acronym entries, citations, references to sections and floats, etc. The parameter `pdfpagelabels` allows the definition of custom page labels, such as `\CoverName` (line 588) for the first page, roman numerals for the following few pages, and arabic numerals for the rest of the document. Without that parameter, all page labels would be arabic numerals from one to the total number of pages. The `url` package doesn't need to be loaded independently since it is called by `hyperref`. For more information about the package, consult its CTAN documentation [28].

The `\hypersetup` macro defines `hyperref`'s global parameters. In our case:

- `bookmarksnumbered`: add numbered bookmarks when browsing the document tree on PDF viewers;
- `bookmarksopen`: sets bookmarks in the document tree as opened by default when reading the PDF;
- `bookmarksopenlevel`: controls the number of section levels that are open in the document tree. In this case, we only want as many levels as are displayed in the ToC (see § 5.9);
- `bookmarksdepth`: controls the number of section levels that receive a bookmark for the PDF viewer. In this case, we want as many bookmark levels as there are section levels;
- `pdfhighlight`: sets the behavior when clicking on a hyperlink inside the PDF. Argument `/N` removes any effect. If the hyperlink works, effects are not necessary;
- `colorlinks`: allows the coloring of hyperlinks;
- `urlcolor`: sets the color for URLs;
- `linkcolor`: sets the color for general hyperlinks (sections, glossary entries, acronym entries, etc.);
- `citecolor`: sets the color for citations;
- `linktoc`: define what part of a ToC entry has hyperlink functionality.

The `backref` package, which is commented out at the moment, provides back referencing from bibliographical citations (indicate on which pages a reference has been cited).

5.19 COUNTERS AND NUMBERING

```
652 \AtBeginDocument{\counterwithin*{table}{section}}
653 \AtBeginDocument{\counterwithin*{figure}{section}}
654 \AtBeginDocument{\counterwithin*{equation}{section}}
655 \AtBeginDocument{\counterwithin*{lstlisting}{section}}
656
657 \AtBeginDocument{\renewcommand{\thetable}{\thesection.\arabic{table}}}
658 \AtBeginDocument{\renewcommand{\thefigure}{\thesection.\arabic{figure}}}
659 \AtBeginDocument{\renewcommand{\thesubfigure}{(\alph{subfigure})}}
660 \AtBeginDocument{\renewcommand{\theequation}{\thesection.\arabic{equation}}}
661 \AtBeginDocument{\renewcommand{\thelstlisting}{\thesection.\arabic{lstlisting}}}
```

Listing 5.23 Counters and numbering settings.

The first four lines reset the way counters are incremented per the requirement of the Université de Sherbrooke: equations, figures, listings, and tables must be counted within a section.

The last five lines define the appearance of the floating environment bookmarks. Most use the section bookmark followed by their own counter value, except for subfigures that simply use lowercase letters.

5.20 SECTIONS

```
667 \usepackage{titlesec}
668
669 \renewcommand{\thesection}{\arabic{section}}
670 \renewcommand{\thesubsection}{\thesection.\arabic{subsection}}
671 \renewcommand{\thesubsubsection}{\thesubsection.\arabic{subsubsection}}
672
673 \titleformat{\section}{\normalfont\Large\bfseries\scshape}{\thesection}{1em}{}
674 \titleformat{\subsection}{\normalfont\large\bfseries\scshape}{\thesubsection}{1em}{}
675 \titleformat{\subsubsection}{\normalfont\normalsize\bfseries\scshape}{\thesubsubsection}{1em}{}
676 \titleformat{\paragraph}{\normalfont\normalsize\bfseries}{\theparagraph}{1em}{}
677
678 \titlespacing*{\section}{0pt}{24pt}{12pt}
679 \titlespacing*{\subsection}{0pt}{12pt}{12pt}
680 \titlespacing*{\subsubsection}{0pt}{12pt}{6pt}
681 \titlespacing*{\paragraph}{0pt}{3.25ex plus 1ex minus .2ex}{1.5ex plus .2ex}
```

Listing 5.24 Sections settings.

The `titlesec` package can be used to customize the appearance of section titles and bookmarks. The first set of lines (669–671) is used to change the appearance of numbered section bookmarks. The next set (673–676) redefines the section title formatting by modifying the font and the spacing after the bookmark. The last set (678–681) specifies the left margin and the vertical

spacing around section titles.

5.21 LISTS

```
687 \usepackage{enumitem}
688
689 % \newcommand{\newbullet}{
690 %     \raisebox{1.5pt}{\tiny$\bullet$}
691 % }
692 % \newcommand{\myitemize}{
693 %     \renewcommand\labelitemi{$\newbullet$}
694 % }
695
696 \setlist[1]{leftmargin=2em}
697 \setlist[2]{leftmargin=2em} % nested list
698 \setlist[3]{leftmargin=2em} % double nested list
699 \setlist[4]{leftmargin=2em} % triple nested list
700 \setlist[itemize,1]{itemsep=0pt,topsep=0pt,label={--}}
701 \setlist[itemize,2]{itemsep=0pt,topsep=0pt,label={\textbullet}}
702 \setlist[itemize,3]{itemsep=0pt,topsep=0pt,label={--}}
703 \setlist[itemize,4]{itemsep=0pt,topsep=0pt,label={\textbullet}}
704 % \setlist[enumerate,1]{itemsep=0pt,topsep=0pt,label={\arabic*}}
705 % \setlist[enumerate,2]{itemsep=0pt,topsep=0pt,label={\alph*}}
```

Listing 5.25 Lists settings.

The `enumitem` package controls the layout of list environments, such as `itemize`, `enumerate`, and `description`. We call it to modify the spacing between the item macro the following content (lines 696–699) as well as changing the bullet points used (lines 700–705). The custom command `\newbullet` and the custom environment `myitemize` are commented out since it’s not necessary in our case to have a personalized bullet point.

5.22 GLOSSARY

```
14 % Glossaries, because they don't play nice when moved
15 \usepackage[acronym,nopostdot,toc,numberline]{glossaries}
16 % \renewcommand*{\glstextformat}[1]{\textcolor{ForestGreen}{#1}}
17 % \renewcommand*{\glsselist}[1]{\textcolor{ForestGreen}{#1}}
18 \makeglossaries
19 \loadglsentries{tools/acronyms}
20 \loadglsentries{tools/glossary}
```

Listing 5.26 Glossary settings.

The `glossaries` package is annoying. It cannot work properly if not called directly in your main document file, which hampers flexibility. Nevertheless, it is an awesome package to manage a glossary and a list of acronyms. It is loaded with a couple parameters:

- acronym: sets an associated conditional to determine whether or not to define a separate list for acronyms;
- nopostdot: suppresses the post-description dot;
- toc: sets the glossary and list of acronyms to appear in the ToC;
- numberline: adds an indentation for the list of acronym and glossary titles in the ToC, same as the space for numbered bookmarks.

Then, the `\glstextformat` and `\glseelist` macros are renewed to change the hyperlink color. This way, it stands out from the generic hyperlinks. The other macros are there to create the files necessary for glossary compilation and to load the entries.

5.22.1 GLOSSARY STYLE

```

116 \newglossarystyle{glscustom}{%
117   \renewenvironment{theglossary}{%
118     \renewcommand{\glsgroupskip}{}
119     \setlength{\tabcolsep}{0pt}
120     \setlength{\columnA}{\widthof{\textbf{Environment}}}%
121     \setlength{\columnB}{\dimexpr 0.02\linewidth}%
122     \setlength{\columnD}{\dimexpr 2\glspagelistwidth}%
123     \setlength{\columnC}{\dimexpr \linewidth-\columnA-2\columnB-\columnD}%
124
125     \setlength{\columnA}{\columnA-2\tabcolsep}
126     \setlength{\columnB}{\columnB-2\tabcolsep}
127     \setlength{\columnC}{\columnC-2\tabcolsep}
128     \setlength{\columnD}{\columnD-2\tabcolsep}
129
130     \begin{longtable*}{%
131       \RT{\columnA}%
132       \LT{\columnB}%
133       \LT{\columnC}%
134       \LT{\columnB}%
135       \LT{\columnD}%
136     }%
137   }\end{longtable*}}%
138
139 \renewcommand*{\glossaryheader}{}%
140 \renewcommand{\glossentry}[2]{%
141   \glstryitem{##1}\glstarget{##1}{\bfseries\glossentryname{##1}}
142   &&\parbox[t]{\linewidth}{\Glossentrydesc{##1}}
143   &&##2\tabularnewline
144 }%
145
146 \renewcommand{\subglossentry}[3]{%
147   &&\glssubentryitem{##2}\glstarget{##2}{\strut}\Glossentrydesc{##2}
148   &&##3\tabularnewline
149 }%
```


Listing 5.27 Glossary style.

The `\newglossarystyle{ }{ }` macro is used to create a custom layout for a glossary or a list of acronyms. In this case, I’ve defined a new layout for acronyms. The standard `long3col` style and its derivatives lacked a couple features that I’ve implemented in this one, such as:

- the table width, which is equal to the text area;
- the cell padding has been deleted so the longest acronym sits flush with the left margin;
- acronyms have been right-aligned and their series has been set to bold;
- space for page references has been doubled;
- extra vertical space between acronyms that start with different letters has been suppressed;
- descriptions always start with a capitalized letter.

For more info on tables in general, see § 2.9.

5.23 CUSTOM PACKAGES

```

711 \usepackage{local_basics}
712 % \usepackage{local_conversions}
713 \usepackage{local_kinematics}
714 % \usepackage{local_physics}

```

Listing 5.28 Custom packages.

Custom packages regroup custom macros by theme. Admittedly, I am not defining my packages in a proper T_EXnician way and would need to better split packages from general settings. Nonetheless, this template include at the moment four packages:

- `local_basics`, which include many mathematical commands and other utilities;
- `local_conversions`, which include functions to convert between units. This was more of a L^AT_EX experiment than a truly useful package for me;
- `local_kinematics`, which defines more mathematical commands for some of the Université de Sherbrooke’s courses;
- `local_physics`, which defines mathematical commands for physics.

5.24 MISCELLANEOUS

```

593 %%% Input files from other directories
594 \usepackage{import}
595
596 %%% Comment environment
597 % \usepackage{comment}
598
599 %%% Character map table for search compatibility
600 %%% in PDF reader (PDFLaTeX only).
601 % \usepackage{cmap}
602
603 %%% Object stacking
604 % \usepackage{stackengine}
605
606 %%% Key-value decoder
607 % \usepackage{keyval}
608
609 %%% Spacing after macro if followed by specific punctuation
610 % \usepackage{xspace}
611
612 %%% Macros with many optional arguments
613 % \usepackage{xargs}
614
615 %%% Reset / unset counters
616 % \usepackage{chngcntr}
617
618 %%% Add background picture
619 \usepackage{eso-pic}
620
621 %%% Add notes for teamwork
622 % \usepackage[colorinlistoftodos]{todonotes}

```

Listing 5.29 Miscellaneous settings.

Other useful packages we could use are:

- `import`: allows file input from relative paths;
- `comment`: creates an environment to selectively include or exclude text from the document;
- `cmap`: makes PDF files searchable and copyable. This is natively supported for engines $\text{X}\text{\LaTeX}$ and $\text{Lua}\text{\LaTeX}$, and thus only necessary with $\text{pdf}\text{\LaTeX}$ (see § 1.1);
- `stackengine`: provides functionalities for stacking objects vertically;
- `keyval`: processes key-value pairs inside package and command arguments (already called by the `graphicx` package);
- `xspace`: provides automatic insertion of space when necessary. When defining a command with `\xspace` at the end, you don’t have to include closing braces to make sure there will be space after the command is called;
- `xargs`: provides extended versions of `\newcommand{ } { }` which allows for easy and robust

definition of macros with optional parameters;

- `chngcnt`: defines commands to reset the way counters are incremented;
- `eso-pic`: provides utilities to add background pictures to a document;
- `todonotes`: helps mark things to do inside the PDF, so they're not missed when writing the document. Useful for collaborative projects, although Overleaf provides its own utility for writing down comments.

6 TEMPLATES

6.1 EQUATIONS

Single equation:

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \left. \frac{d^n f}{dx^n} \right|_{x=x_0} (x - x_0)^n \quad (6.1)$$

```

1 \begin{equation}
2   f\inp{x}
3   = \sum\limits_{n=0}^{\infty} \frac{1}{n!}
4   \left.\deriv{n}{f}{x}\right|_{x=x_0} \inp{x - x_0}^n
5   \label{eq:mylabel_01}
6 \end{equation}

```

Listing 6.1 Template: single equation.

Set of equations:

$$\begin{aligned} \sin(\alpha \pm \beta) &= \sin(\alpha) \cos(\beta) \pm \cos(\alpha) \sin(\beta) \\ \cos(\alpha \pm \beta) &= \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta) \\ \tan(\alpha \pm \beta) &= \frac{\tan(\alpha) \pm \tan(\beta)}{1 \mp \tan(\alpha) \tan(\beta)} \end{aligned} \quad (6.2)$$

```

1 \begin{equation}
2   \setlength{\jot}{5pt}
3   \begin{aligned}
4     \sin\inp{\alpha\pm\beta}
5     &= \sin\inp{\alpha}\cos\inp{\beta} \pm \cos\inp{\alpha}\sin\inp{\beta}
6     \\
7     \cos\inp{\alpha\pm\beta}
8     &= \cos\inp{\alpha}\cos\inp{\beta} \mp \sin\inp{\alpha}\sin\inp{\beta}
9     \\
10    \tan\inp{\alpha\pm\beta}
11    &= \frac{\tan\inp{\alpha} \pm \tan\inp{\beta}}{1 \mp \tan\inp{\alpha}\tan\inp{\beta}}
12  \end{aligned}
13  \label{eq:mylabel_02}
14 \end{equation}

```

Listing 6.2 Template: set of equations.

Mathematical development:

$$\begin{aligned}
\frac{Q}{C} &= \frac{dT}{dt} + \frac{T}{RC} \\
\Rightarrow \mathcal{L}\left(\frac{Q}{C}\right) &= \mathcal{L}\left(\frac{dT}{dt} + \frac{T}{RC}\right) \\
\Rightarrow \frac{Q}{C} &= sT(s) + \frac{T(s)}{RC} \\
\Rightarrow T(s) &= \frac{Q/C}{s + 1/RC} \\
\Rightarrow T(s) &= RQ \left(\frac{1}{RCs + 1} \right)
\end{aligned} \tag{6.3}$$

```

1  {
2  \setlength{\jot}{10pt}
3  \begin{align}
4    \frac{Q}{C}
5    &= \frac{dT}{dt} + \frac{T}{RC}
6    \nonumber\\
7    &\Rightarrow \mathcal{L}\left(\frac{Q}{C}\right)
8    = \mathcal{L}\left(\frac{dT}{dt} + \frac{T}{RC}\right)
9    \nonumber\\
10   &\Rightarrow \frac{Q}{C}
11   = sT(s) + \frac{T(s)}{RC}
12   \nonumber\\
13   &\Rightarrow T(s)
14   = \frac{\frac{Q}{C}}{s + \frac{1}{RC}}
15   \nonumber\\
16   &\Rightarrow T(s)
17   = RQ \frac{1}{RCs + 1}
18   \label{eq:mylabel_03}
19 \end{align}
20 }

```

Listing 6.3 Template: mathematical development.

6.2 LISTS

- First level:
 - Second level:
 - Third level:
 - Fourth level (I);
 - Fourth level (II).

–

•

–

```

1  \begin{itemize}
2      \item First level:
3      \begin{itemize}
4          \item Second level:
5          \begin{itemize}
6              \item Third level:
7              \begin{itemize}
8                  \item Fourth level (I);
9                  \item Fourth level (II).
10             \end{itemize}
11             \item
12         \end{itemize}
13         \item
14     \end{itemize}
15     \item
16 \end{itemize}

```

Listing 6.4 Template: unordered list.

1. First level:
 - (a) Second level:
 - i. Third level:
 - A. Fourth level (I);
 - B. Fourth level (II).
 - ii.
 - (b)
- 2.

```

1 \begin{enumerate}
2   \item First level:
3   \begin{enumerate}
4     \item Second level:
5     \begin{enumerate}
6       \item Third level:
7       \begin{enumerate}
8         \item Fourth level (I);
9         \item Fourth level (II).
10      \end{enumerate}
11     \item
12   \end{enumerate}
13   \item
14 \end{enumerate}
15 \item
16 \end{enumerate}

```

Listing 6.5 Template: ordered list.

6.3 FIGURES



Figure 6.1 Caption under figure.

```

1 \begin{figure}[H]
2   \centering
3   \includegraphics[width=\figsize\columnwidth]{data/udes/udes.pdf}
4   \caption[Template: single figure]{Caption under figure.}
5   \label{fig:figlabel_01}
6 \end{figure}

```

Listing 6.6 Template: single figure.

```

1 \begin{figure}[H]
2   \centering
3   \caption[Template: single figure, top caption]{Caption over figure.}
4   \includegraphics[width=\figsize\columnwidth]{data/udes/udes.pdf}
5   \label{fig:figlabel_01}
6 \end{figure}

```

Listing 6.7 Template: single figure, top caption.

(a) Caption under subfigure.

(b) Caption under subfigure.

Figure 6.2 Caption under figure.

```

1 \begin{figure}[H]
2   \centering
3   \begin{subfigure}[t]{0.5\textwidth}
4     \centering
5     \includegraphics[width=\figsize\columnwidth]{data/udes/3it.pdf}
6     \caption{Caption under subfigure.}
7     \label{fig:figsublabel_01}
8   \end{subfigure}%
9   \begin{subfigure}[t]{0.5\textwidth}
10    \centering
11    \includegraphics[width=\figsize\columnwidth]{data/udes/udes.pdf}
12    \caption{Caption under subfigure.}
13    \label{fig:figsublabel_02}
14  \end{subfigure}
15  \caption[Template: figure with subfigures]{Caption under figure.}
16  \label{fig:figlabel_02}
17 \end{figure}

```

Listing 6.8 Template: figure with subfigures.

6.4 TABLES

Table 6.1 Caption over table.

Column title #1	Column title #2	Column title #3
Cell #1-1	Cell #2-1	Cell #3-1
Cell #1-2	Cell #2-2	Cell #3-2
Cell #1-3	Cell #2-3	Cell #3-3

```

1 % Define column widths
2 \setlength{\columnA}{\linewidth/3}
3 \setlength{\columnB}{\linewidth/3}
4 \setlength{\columnC}{\linewidth-\columnA-\columnB}
5

```



```

6      % Subtract column padding and vertical line thickness
7      \setlength{\columnA}{\columnA-2\tabcolsep-4\vbar/3}
8      \setlength{\columnB}{\columnB-2\tabcolsep-4\vbar/3}
9      \setlength{\columnC}{\columnC-2\tabcolsep-4\vbar/3}
10
11     \begin{longtable}%
12         {\RC{\columnA}|% Right + center alignment
13          \CC{\columnB}|% Center + center alignment
14          \LC{\columnC}|% Left + center alignment
15         }
16     \caption[Template: simple table]{Caption over table.}%
17     \label{tab:mylabel_01}\\
18
19     % First header
20     \hline
21     \multicolumn{1}{|\CC{\columnA}|}{Column title \#1}
22         & Column title \#2
23         & \multicolumn{1}{|\CC{\columnC}|}{Column title \#3}
24     \\ \hline
25     \endfirsthead
26
27     % Main header
28     \hline
29     \multicolumn{1}{|\CC{\columnA}|}{Column title \#1}
30         & Column title \#2
31         & \multicolumn{1}{|\CC{\columnC}|}{Column title \#3}
32     \\ \hline
33     \endhead
34
35     % First row
36     Cell \#1-1
37         & Cell \#2-1
38         & Cell \#3-1
39     \\ \hline
40
41     % Second row
42     Cell \#1-2
43         & Cell \#2-2
44         & Cell \#3-2
45     \\ \hline
46
47     % Third row
48     Cell \#1-3
49         & Cell \#2-3
50         & Cell \#3-3
51     \\ \hline
52     \end{longtable}
53 \endgroup

```

Listing 6.9 Template: simple table.

Table 6.2 Caption over table.

	Column title #1	Column title #2	
		Column title #3	Column title #4
Cell #1	Cell #2-1	Cell #3-1	Cell #4-1
	Cell #2-2	Cell #3-2	Cell #4-2
	Cell #2-3	Cell #3-3	Cell #4-3

```

1 \begin{group
2   % Define column widths
3   \setlength{\columnA}{\linewidth/4}
4   \setlength{\columnB}{\linewidth/4}
5   \setlength{\columnC}{\linewidth/4}
6   \setlength{\columnD}{\linewidth-\columnA-\columnB-\columnC}
7
8   % Subtract column padding and vertical line thickness
9   \setlength{\columnA}{\columnA-2\tabcolsep-5\vbar/4}
10  \setlength{\columnB}{\columnB-2\tabcolsep-5\vbar/4}
11  \setlength{\columnC}{\columnC-2\tabcolsep-5\vbar/4}
12  \setlength{\columnD}{\columnD-2\tabcolsep-5\vbar/4}
13
14  \begin{longtable}%
15    {\LT{\columnA}}| % Left + top alignment
16    \LT{\columnB}}| % Left + top alignment
17    \LT{\columnC}}| % Left + top alignment
18    \LT{\columnD}}| % Left + top alignment
19  }
20  \caption[Template: table with merged rows and columns]{Caption over table.}%
21  \label{tab:mylabel_02}\\
22
23  % First header
24  \cline{2-4}
25  \multicolumn{1}{c|}{~}
26    &\multirow{2}{\linewidth}{\centering Column title \#1}
27    &\multicolumn{2}{\CC{\columnC+\columnD+2\tabcolsep+\vbar}}{|}{Column title \#2}
28  \\ \cline{3-4}
29  \multicolumn{1}{c|}{~}
30    &&\multicolumn{1}{\CC{\columnC}}{|}{Column title \#3}
31    &\multicolumn{1}{\CC{\columnD}}{|}{Column title \#4}
32  \\ \cline{2-4}
33  \endfirsthead
34
35  % Main header
36  \cline{2-4}
37  \multicolumn{1}{c|}{~}
38    &\multirow{2}{\linewidth}{\centering Column title \#1}

```

```

39      &\multicolumn{2}{\CC{\columnC+\columnD+2\tabcolsep+\vbar}}{Column title \#2}
40    \\ \cline{3-4}
41    \multicolumn{1}{c|}{~}
42      &&\multicolumn{1}{\CC{\columnC}}{Column title \#3}
43      &\multicolumn{1}{\CC{\columnD}}{Column title \#4}
44    \\ \cline{2-4}
45    \endhead
46
47    % First row
48    \multirow{3}{\linewidth}{\centering Cell \#1}
49      & Cell \#2-1
50      & Cell \#3-1
51      & Cell \#4-1
52    \\ \cline{2-4}
53
54    % Second row
55      & Cell \#2-2
56      & Cell \#3-2
57      & Cell \#4-2
58    \\ \cline{2-4}
59
60    % Third row
61      & Cell \#2-3
62      & Cell \#3-3
63      & Cell \#4-3
64    \\ \hline
65    \end{longtable}
66  \endgroup

```

Listing 6.10 Template: table with merged rows and columns.

Table 6.3 Caption over table.

Column title #1	Column title #2	Column title #3
Cell #1-1	Cell #2-1	Cell #3-1
Cell #1-2	Cell #2-2	Cell #3-2
Cell #1-3	Cell #2-3	Cell #3-3

```

1  % Landscape geometry
2  \loadgeometry{landscape}
3
4  \begin{group
5      % Reduce font size (n / n+1)
6      \large
7
8      % Define column widths
9      \setlength{\columnA}{\linewidth/2}
10     \setlength{\columnB}{\linewidth/3}
11     \setlength{\columnC}{\linewidth-\columnA-\columnB}
12
13     % Subtract column padding and vertical line thickness
14     \setlength{\columnA}{\columnA-2\tabcolsep-4\vbar/3}
15     \setlength{\columnB}{\columnB-2\tabcolsep-4\vbar/3}
16     \setlength{\columnC}{\columnC-2\tabcolsep-4\vbar/3}
17
18     \begin{longtable}%
19         {\RC{\columnA}|% Right + center alignment
20          \CC{\columnB}|% Center + center alignment
21          \LC{\columnC}|% Left + center alignment
22         }
23     \caption[Template: simple table]{Caption over table.}%
24     \label{tab:mylabel_03}\\
25
26     % First header
27     \hline
28     \multicolumn{1}{|\CC{\columnA}|}{Column title \#1}
29         & Column title \#2
30         & \multicolumn{1}{|\CC{\columnC}|}{Column title \#3}
31     \\ \hline
32     \endfirsthead
33
34     % Main header
35     \hline
36     \multicolumn{1}{|\CC{\columnA}|}{Column title \#1}
37         & Column title \#2
38         & \multicolumn{1}{|\CC{\columnC}|}{Column title \#3}
39     \\ \hline
40     \endhead
41
42     % First row
43     Cell \#1-1
44         & Cell \#2-1
45         & Cell \#3-1
46     \\ \hline
47
48     % Second row
49     Cell \#1-2
50         & Cell \#2-2
51         & Cell \#3-2
52     \\ \hline
53
54     % Third row
55     Cell \#1-3
56         & Cell \#2-3
57         & Cell \#3-3
58     \\ \hline
59     \end{longtable}
60 \endgroup
61
62 % Portrait geometry
63 \restoregeometry

```

Listing 6.11 Template: landscape table.

Table 6.4 Caption over table.

Column title #1	Column title #2	Column title #3
Cell #1-1	Cell #2-1	– Cell #3-1 (I); – Cell #3-1 (II).
Cell #1-2	Cell #2-2	– Cell #3-2 (I); – Cell #3-2 (II).
Cell #1-3	Cell #2-3	– Cell #3-3 (I): • A; • B; • C. – Cell #3-3 (II).

```

1 \begin{group
2   % Define column widths
3   \setlength{\columnA}{\linewidth/3}
4   \setlength{\columnB}{\linewidth/3}
5   \setlength{\columnC}{\linewidth-\columnA-\columnB}
6
7   % Subtract column padding and vertical line thickness
8   \setlength{\columnA}{\columnA-2\tabcolsep-4\vbar/3}
9   \setlength{\columnB}{\columnB-2\tabcolsep-4\vbar/3}
10  \setlength{\columnC}{\columnC-2\tabcolsep-4\vbar/3}
11
12  \begin{longtable}%
13    {\RB{\columnA}}| % Right + bottom alignment
14    \CB{\columnB}}| % Center + bottom alignment
15    \LB{\columnC}}| % Left + bottom alignment
16  }
17  \caption[Template: table with multi-line cells and lists]{Caption over table.}%
18  \label{tab:mylabel_04}\\
19
20  % First header
21  \hline
22  \multicolumn{1}{|\CC{\columnA}}|{Column title \#1}
23    & Column title \#2
24    & \multicolumn{1}{|\CC{\columnC}}|{Column title \#3}
25  \\ \hline
26  \endfirsthead
27
28  % Main header
29  \hline
30  \multicolumn{1}{|\CC{\columnA}}|{Column title \#1}
31    & Column title \#2

```

```

32         & \multicolumn{1}{\CC{\columnC}}{Column title \#3}
33     \\ \hline
34     \endhead
35
36     % First row
37     Cell \#1-1
38         & Cell \#2-1
39         & -- Cell \#3-1 (I)
40         \newline -- Cell \#3-1 (II)
41     \\ \hline
42
43     % Second row
44     Cell \#1-2
45         & Cell \#2-2
46         & -- Cell \#3-2 (I)
47         \newline -- Cell \#3-2 (II)
48     \\ \hline
49
50     % Third row
51     Cell \#1-3
52         & Cell \#2-3
53         & -- Cell \#3-3 (I)
54         \newline \quad \tabitem A
55         \newline \quad \tabitem B
56         \newline \quad \tabitem C
57         \newline -- Cell \#3-3 (II)
58     \\ \hline
59     \end{longtable}
60 \endgroup

```

Listing 6.12 Template: table with multi-line cells and lists.

6.5 TikZ

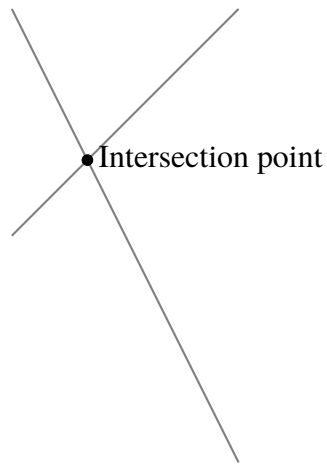
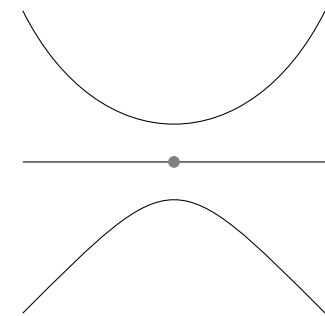
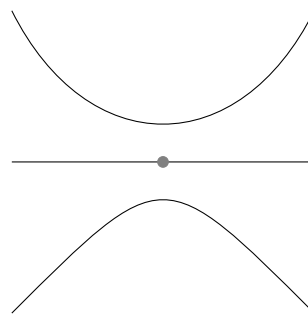


Figure 6.3 Caption under figure.



(a) Caption under subfigure.



(b) Caption under subfigure.

Figure 6.4 Caption under figure.

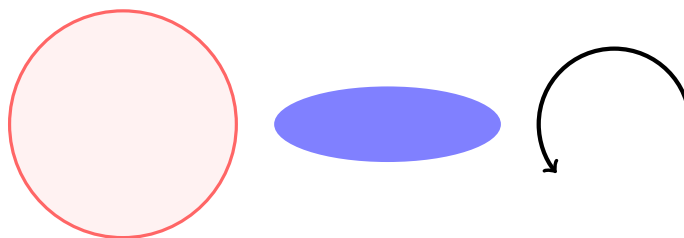


Figure 6.5 Caption under figure.

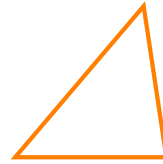


Figure 6.6 Caption under figure.

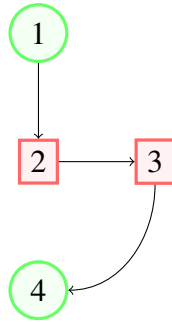


Figure 6.7 Caption under figure.

Table 6.5 Caption over table.

TMF Tasks			
Task	Description	Starting Date	End Date
Project Start	First meeting at which the project was planned (its structure and the objectives to be achieved at the end of the project).	30/11/18	30/11/18
Theoretical Analysis	Initial study of the system based on the documentation	01/12/18	23/12/18
Matlab simulations			
Development of equations	The first equations were developed in Matlab, verifying that the results from the University of Rochester [REF] were obtained for the cases of 3 and 4 lenses. Later, the system was generalized for the case of 5 lenses, obtaining the first problems.	24/12/18	10/01/19
Meeting	To comment on the progress made and try to solve problems in simulations with Matlab.	31/01/19	31/01/19
Meeting	To compare the Matlab equations and try to find the solution.	06/03/19	06/03/19
Meeting	Finally, we fixed the problem with the Matlab equations.	08/04/19	08/04/19
Zemax simulations			
Zemax installation	We had some problems during the installation of the program due to the Windows version of the computer.	01/01/19	03/03/19
First steps with Zemax	First days learning to use the program.	25/03/19	30/03/19
Meeting	Skype meeting to clarify some doubts regarding Zemax simulations.	27/04/19	27/04/19

$$\mathbf{X} = \begin{array}{c} \begin{array}{c} B' \\ B'' \end{array} \left\{ \begin{array}{cc} \overbrace{\begin{array}{cc} v_{1,1} & v_{1,2} \end{array}}^{C'} & \overbrace{\begin{array}{cc} v_{1,3} & \end{array}}^{C''} \\ \begin{array}{cc} v_{2,1} & v_{2,2} \end{array} & v_{2,3} \\ \begin{array}{cc} v_{3,1} & v_{3,2} \end{array} & \begin{array}{c} v_{3,3} \\ v_{4,3} \end{array} & \begin{array}{c} v_{3,4} \\ v_{4,4} \end{array} & \begin{array}{cc} v_{4,5} & v_{4,6} \end{array} \\ \begin{array}{c} v_{5,4} \\ v_{6,4} \end{array} & \begin{array}{cc} v_{5,5} & v_{5,6} \\ v_{6,5} & v_{6,6} \end{array} \end{array} \right\} \begin{array}{c} E' \\ E'' \end{array} \end{array} \quad (6.4)$$

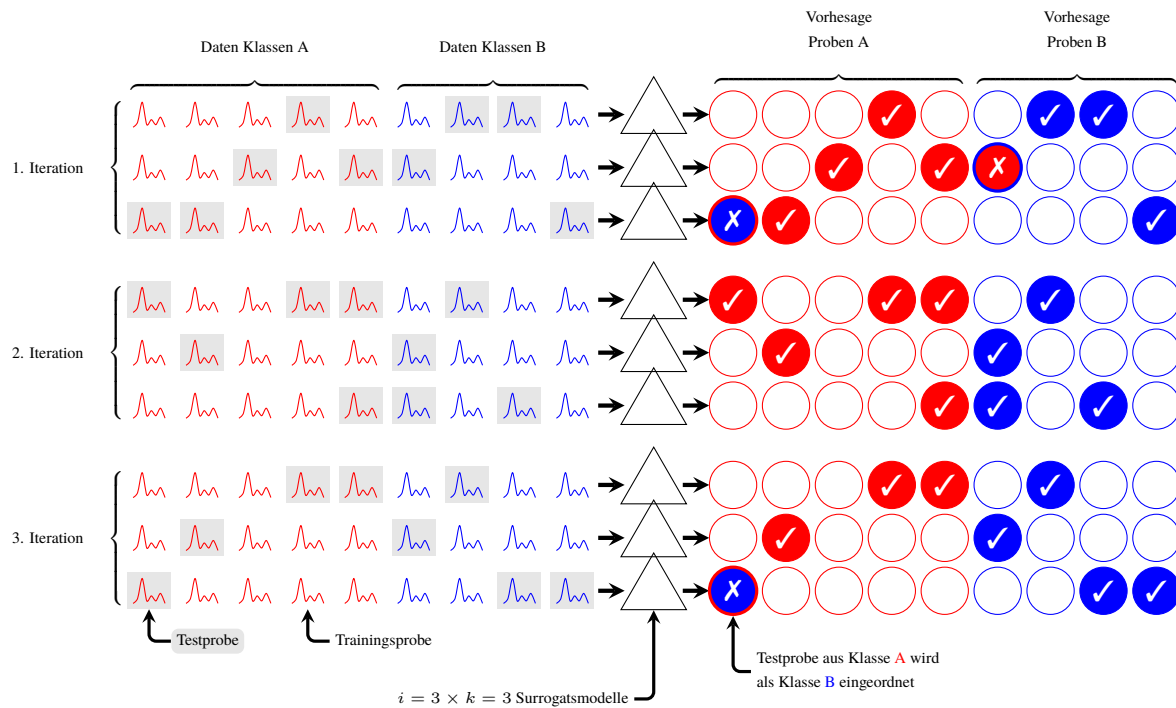


Figure 6.8 Caption under figure.

Table 6.6 Caption over table.

	Monday	Tuesday	Wednesday	Thursday	Friday
1	A	B	C	D	E
2	F	G	H	J	K
3	A	B	C	D	E
4	F	G	H	J	K

Table 6.7 Caption over table.

8	1	6
3	5	7
4	9	2

6.6 BibTeX

```
1 @article{template:article,
2   %%% Required
3   author      = {Author},
4   title       = {Article title},
5   journaltitle = {Periodical title},
6   % year      = {}, % Choose if you only know the year
7   % month     = {}, % Pair with 'year' if you don't know the day
8   date        = {2000-01-01}, % Choose if you know the entire date
9   pages       = {\#\#--\#\#}, % Choose for single page or range of pages
10  % pagetotal  = {}, % Choose for page total
11
12  %%% Provide if exists
13  location     = {Location},
14  volume       = {5},
15  number       = {Number},
16
17  %%% Suggested (pick at least one type)
18  doi          = {10.1109/AERO.2014.6836308}, % Digital Object Identifier
19  % eid        = {}, % Electronic Identifier
20  % eprint     = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
21  % eprinttype = {}, % Electronic publishing type (ex.: arxiv)
22  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
23  % isbn       = {}, % International Standard Book Number
24  % issn       = {}, % International Standard Serial Number
25  % url        = {},
26
27  %%% Utilities
28  % crossref   = {}, % Inherit fields from other reference
29  % keywords   = {keyword_01, keyword_02} % For filtered bibliographies
30 }
```

Listing 6.13 Template: article source.

```

1 @inproceedings{template:inproceedings,
2   %%% Required
3   author      = {Author},
4   title       = {Article title},
5   booktitle   = {Proceedings title},
6   publisher   = {Publishing house},
7   % year      = {}, % Choose if you only know the year
8   % month     = {}, % Pair with 'year' if you don't know the day
9   date        = {2000-01-01}, % Choose if you know the entire date
10  pages       = {\#\#--\#\#}, % Choose for single page or range of pages
11  % pagetotal  = {}, % Choose for page total
12
13  %%% Provide if exists
14  editor       = {Editor},
15  organisation = {Organisation},
16  series       = {Collection},
17  location     = {Location},
18  volume       = {5},
19
20  %%% Suggested (pick at least one type)
21  % doi        = {}, % Digital Object Identifier
22  % eid        = {}, % Electronic Identifier
23  eprint       = {1008.2849v1}, % Electronic publishing identifier
24  eprinttype   = {arxiv}, % Electronic publishing type
25  eprintclass  = {cs.DS}, % Electronic publishing class
26  % isbn       = {}, % International Standard Book Number
27  % issn       = {}, % International Standard Serial Number
28  % url        = {},
29
30  %%% Utilities
31  % crossref   = {}, % Inherit fields from other reference
32  % keywords   = {keyword_01, keyword_02} % For filtered bibliographies
33 }

```

Listing 6.14 Template: in proceedings source.

```

1 @book{template:book,
2   %%% Required
3   author      = {Author},
4   % editor     = {Editor}, % Choose if author unknown
5   title       = {Book title},
6   year        = {2000},
7   edition     = {2},
8   publisher    = {Publishing house},
9   % pages      = {}, % Choose for single page or range of pages
10  pagetotal    = {420}, % Choose for page total
11
12  %%% Provide if exists
13  location     = {Location},
14  volume       = {5},
15
16  %%% Suggested (pick at least one type)
17  % doi         = {}, % Digital Object Identifier
18  % eid         = {}, % Electronic Identifier
19  % eprint      = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
20  % eprinttype  = {}, % Electronic publishing type (ex.: arxiv)
21  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
22  isbn         = {978-1-4799-1622-1}, % International Standard Book Number
23  % issn        = {}, % International Standard Serial Number
24  % url         = {},
25
26  %%% Utilities
27  % crossref    = {}, % Inherit fields from other reference
28  % keywords    = {keyword_01, keyword_02} % For filtered bibliographies
29 }

```

Listing 6.15 Template: book source.

```

1 @inbook{template:inbook,
2   %%% Required
3   author      = {Author},
4   % editor     = {Editor}, % Choose if author unknown
5   title       = {Part title},
6   booktitle   = {Book title},
7   year        = {2000},
8   edition     = {2},
9   publisher    = {Publishing house},
10  % pages       = {}, % Choose for single page or range of pages
11  pagetotal    = {420}, % Choose for page total
12
13  %%% Provide if exists
14  location     = {Location},
15  volume       = {5},
16
17  %%% Suggested (pick at least one type)
18  % doi         = {}, % Digital Object Identifier
19  % eid         = {}, % Electronic Identifier
20  % eprint      = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
21  % eprinttype  = {}, % Electronic publishing type (ex.: arxiv)
22  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
23  isbn         = {978-1-4799-1622-1}, % International Standard Book Number
24  % issn        = {}, % International Standard Serial Number
25  % url         = {},
26
27  %%% Utilities
28  % crossref    = {}, % Inherit fields from other reference
29  % keywords    = {keyword_01, keyword_02} % For filtered bibliographies
30 }

```

Listing 6.16 Template: in book source.


```

1 @thesis{template:thesis,
2   %%% Required
3   author      = {Author},
4   title       = {Title},
5   year        = {2000},
6   type        = {Degree level},
7   institution = {Institution},
8   location    = {Location},
9   pages       = {\#\#--\#\#}, % Choose for single page or range of pages
10  % pagetotal  = {}, % Choose for page total
11
12  %%% Suggested (pick at least one type)
13  doi          = {10.1109/AERO.2014.6836308}, % Digital Object Identifier
14  % eid        = {}, % Electronic Identifier
15  % eprint     = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
16  % eprinttype = {}, % Electronic publishing type (ex.: arxiv)
17  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
18  % isbn       = {}, % International Standard Book Number
19  % issn       = {}, % International Standard Serial Number
20  % url        = {},
21
22  %%% Utilities
23  % crossref   = {}, % Inherit fields from other reference
24  % keywords   = {keyword_01, keyword_02} % For filtered bibliographies
25 }

```

Listing 6.17 Template: thesis source.

```

1 @report{template:report,
2   %%% Required
3   author      = {Author},
4   title       = {Title},
5   year        = {2000},
6   institution = {Institution},
7   pages       = {\#\#--\#\#}, % Choose for single page or range of pages
8   % pagetotal  = {}, % Choose for page total
9
10  %%% Provide if exists
11  location     = {Location},
12  type         = {Report type},
13  number       = {Number},
14
15  %%% Suggested (pick at least one type)
16  % doi        = {}, % Digital Object Identifier
17  % eid        = {}, % Electronic Identifier
18  % eprint     = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
19  % eprinttype = {}, % Electronic publishing type (ex.: arxiv)
20  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
21  % isbn       = {}, % International Standard Book Number
22  issn        = {0317-8471}, % International Standard Serial Number
23  % url        = {},
24
25  %%% Utilities
26  % crossref   = {}, % Inherit fields from other reference
27  % keywords   = {keyword_01, keyword_02} % For filtered bibliographies
28 }

```

Listing 6.18 Template: report source.

```

1 @www{template:website,
2   %%% Required
3   author      = {IEEE},
4   title       = {IEEE Xplore},
5   year        = {2014}, % Choose if you only know the year
6   url         = {https://ieeexplore.ieee.org/document/6836308},
7   urldate     = {2021-05-15},
8
9   %%% Provide if exists
10  month       = {03}, % Pair with 'year' if you don't know the day
11  % date      = {}, % Choose if you know the entire date
12
13  %%% Utilities
14  % crossref   = {}, % Inherit fields from other reference
15  % keywords   = {keyword_01, keyword_02} % For filtered bibliographies
16 }

```

Listing 6.19 Template: website source.

```

1 @www{template:web-doc,
2   %%% Required
3   author      = {David Klaus and Robert Ocampo and Christine Fanchiang},
4   editor      = {IEEE},
5   title       = {IEEE Xplore}, % Website title
6   titleaddon  = {Spacecraft human-rating: Historical overview and implementation considerations
7     }, % Resource title
8   year        = {2014},
9   url         = {https://ieeexplore.ieee.org/document/6836308},
10  urldate     = {2021-05-15},
11
12  %%% Provide if exists
13  month       = {03}, % Pair with 'year' if you don't know the day
14  % date      = {}, % Choose if you know the entire date
15
16  %%% Suggested (pick at least one type)
17  doi         = {10.1109/AERO.2014.6836308}, % Digital Object Identifier
18  % eid       = {}, % Electronic Identifier
19  % eprint    = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
20  % eprinttype = {}, % Electronic publishing type (ex.: arxiv)
21  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
22  isbn        = {978-1-4799-1622-1}, % International Standard Book Number
23  issn        = {1095-323X}, % International Standard Serial Number
24
25  %%% Utilities
26  % crossref   = {}, % Inherit fields from other reference
27  % keywords   = {keyword_01, keyword_02} % For filtered bibliographies
28 }

```

Listing 6.20 Template: web resource source.

```

1 @misc{template:legislation,
2   %%% Required
3   title   = {Legislation title},
4   type    = {Jurisdiction},
5   note    = {Number},
6
7   %%% Provide if exists
8   year    = {2000},
9
10  %%% Utilities
11  % crossref      = {}, % Inherit fields from other reference
12  % keywords      = {keyword_01, keyword_02} % For filtered bibliographies
13 }

```

Listing 6.21 Template: legislation source.

```

1 @book{template:norm,
2   %%% Required
3   author    = {Author},
4   title     = {Title},
5   year      = {2000},
6   publisher = {Publishing house},
7   pages     = {\#-\#--\#-\#}, % Choose for single page or range of pages
8   % pagetotal = {}, % Choose for page total
9
10  %%% Provide if exists
11  location  = {Location},
12  number    = {Number},
13
14  %%% Suggested (pick at least one type)
15  % doi      = {}, % Digital Object Identifier
16  % eid      = {}, % Electronic Identifier
17  % eprint   = {}, % Electronic publishing identifier (ex.: 1008.2849v1)
18  % eprinttype = {}, % Electronic publishing type (ex.: arxiv)
19  % eprintclass = {}, % Electronic publishing class (ex. cs.DS)
20  % isbn     = {}, % International Standard Book Number
21  % issn     = {}, % International Standard Serial Number
22  % url      = {},
23
24  %%% Utilities
25  % crossref      = {}, % Inherit fields from other reference
26  % keywords      = {keyword_01, keyword_02} % For filtered bibliographies
27 }

```

Listing 6.22 Template: norm source.

6.7 GLOSSARY

```
1 \newglossaryentry{name}{%
2   name      = {name},
3   % plural   = {names},
4   description = {Blah blah blah},
5   % see      = [Glossary:]{other_glossary_entry}
6 }
```

Listing 6.23 Template: glossary entry.

6.8 ACRONYMS

```
1 \newglossaryentry{NAME}{%
2   type      = \acronymtype,
3   name      = {NAME},
4   description = {Acronym Meaning},
5   first     = {acronym meaning (\glsentrytext{NAME})},
6   plural    = {\glsentrytext{NAME}s},
7   firstplural = {acronym meanings (\glsentrytext{NAME}s)}
8   % see     = [Glossary:]{glossary_entry}
9 }
```

Listing 6.24 Template: acronym entry (native language).

```
1 \newglossaryentry{NOM}{%
2   type      = \acronymtype,
3   name      = {NOM},
4   description = {\emph{Définition de l'acronyme}},
5   first     = {\emph{définition de l'acronyme} (\glsentrytext{NOM})},
6   plural    = {\glsentrytext{NOM}},
7   firstplural = {\emph{définitions de l'acronyme} (\glsentrytext{NOM})}
8   % see     = [Glossary:]{glossary_entry}
9 }
```

Listing 6.25 Template: acronym entry (foreign language).

```
1 \newglossaryentry{NOUN}{%
2   type      = \acronymtype,
3   name      = {NOUN},
4   description = {\emph{Acronym Meaning}},
5   first     = {\glsentrydesc{NOUN} (\glsentrytext{NOUN})}
6   % see     = [Glossary:]{glossary_entry}
7 }
```

Listing 6.26 Template: acronym entry (proper noun).

SOURCES

- [1] Overleaf. *What's in a Name: A Guide to the Many Flavours of T_EX*. 2014. URL: https://www.overleaf.com/learn/latex/Articles/What's_in_a_Name:_A_Guide_to_the_Many_Flavours_of_TeX (visited on 2021-06-01).
- [2] Overleaf. *The T_EX family tree: E_T_X, pdfT_EX, X_YT_EX, LuaT_EX and ConT_EXt*. 2014. URL: https://www.overleaf.com/learn/latex/Articles/The_TeX_family_tree:_LaTeX,_pdfTeX,_XeTeX,_LuaTeX_and_ConTeXt (visited on 2021-06-01).
- [3] Donald E. Knuth. *T_EX and METAFONT, New Directions in Typesetting*. Bedford, Massachusetts, USA: American Mathematical Society and Digital Press, 1979. 353 pp. ISBN: 0-932376-02-9.
- [4] Donald E. Knuth. *Computers & Typesetting: The T_EXbook*. Vol. A. Bedford, Massachusetts, USA: American Mathematical Society and Addison-Wesley, 1986. 483 pp. ISBN: 0-201-13448-9.
- [5] Donald E. Knuth. *Computers & Typesetting: The METAFONTbook*. Vol. C. Bedford, Massachusetts, USA: American Mathematical Society and Addison-Wesley, 1986. 361 pp. ISBN: 0-201-13445-4.
- [6] Petr Olšák. *OpT_EX – LuaT_EX format based on Plain T_EX and OPmac*. 2021-06-04. URL: <https://ctan.org/pkg/optex> (visited on 2021-08-23).
- [7] T_EX Users Group. *T_EX Live*. 2021-04. URL: <https://www.tug.org/texlive/> (visited on 2021-06-01).
- [8] Faculté de génie. *Protocole de rédaction aux études supérieures*. Writing guide. Sherbrooke, Québec, Canada: Université de Sherbrooke, 2018. 54 pp. URL: https://usherbrooke.sharepoint.com/:b:/r/sites/fgen/DocOffFacGen/Chapitre%204/%C3%89tudes%20sup%C3%A9rieures/Protocole_Redaction_EtudesSup.pdf?csf=1&web=1&e=PRL830.
- [9] Wikipedia. *BibT_EX*. 2021-04-15. URL: <https://en.wikipedia.org/wiki/BibTeX> (visited on 2021-06-03).
- [10] Clea F. Rees. *TUG's CTAN archive*. BibT_EX Cheat Sheet. Ed. by T_EX Users Group. 2017-06-24. URL: <http://tug.ctan.org/info/biblatex-cheatsheet/biblatex-cheatsheet.pdf> (visited on 2021-08-23).

- [11] Philip Kime. *Bib \LaTeX – Sophisticated Bibliographies in \LaTeX* . 2020-12-31. URL: <https://ctan.org/pkg/biblatex> (visited on 2021-08-23).
- [12] Nicola Talbot. *glossaries – Create glossaries and lists of acronyms*. 2020-03-20. URL: <https://ctan.org/pkg/glossaries> (visited on 2021-08-23).
- [13] Herbert Voß. *Typesetting Mathematics in \LaTeX* . Cambridge, UK: UIT Cambridge Ltd., 2010. 282 pp. ISBN: 978-1-906860-17-2.
- [14] Herbert Voß. *voss-mathmode – A comprehensive review of mathematics in \LaTeX* . 2010-12-14. URL: <https://ctan.org/pkg/voss-mathmode> (visited on 2021-08-23).
- [15] Matthias Heinkenschloss. *Rice CAAM. \LaTeX Mathematical Symbols*. Ed. by Rice University. 2016. URL: <https://www.caam.rice.edu/~heinken/latex/symbols.pdf> (visited on 2021-08-23).
- [16] The \LaTeX Team and David Carlisle. *graphicx – Enhanced support for graphics*. 2020-12-05. URL: <https://ctan.org/pkg/graphicx> (visited on 2021-08-23).
- [17] Herbert Voß. *Typesetting tables with \LaTeX* . Cambridge, UK: UIT Cambridge Ltd., 2011. 230 pp. ISBN: 978-1-906860-25-7.
- [18] Markus Püschel. *ETH Zürich. Small Guide to Making Nice Tables*. Ed. by Carnegie Mellon University. 2010-10-06. URL: <https://people.inf.ethz.ch/markusp/teaching/guides/guide-tables.pdf> (visited on 2021-08-20).
- [19] Herbert Voß. *PSTricks: Graphics and PostScript for $T_{\text{E}}X$ and \LaTeX* . Cambridge, UK: UIT Cambridge Ltd., 2011. 928 pp. ISBN: 978-1-906860-13-4.
- [20] Wikibooks. *\LaTeX /PGF/TikZ*. 2021-05. URL: <https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ> (visited on 2021-06-22).
- [21] The PGF/TikZ Team et al. *pgf – Create PostScript and PDF graphics in $T_{\text{E}}X$* . 2021-05-16. URL: <https://ctan.org/pkg/pgf> (visited on 2021-08-23).
- [22] Overleaf. *How to use latexmkrc with Overleaf: examples and techniques*. 2014. URL: https://www.overleaf.com/learn/latex/Articles/How_to_use_latexmkrc_with_Overleaf:_examples_and_techniques (visited on 2021-06-01).
- [23] Comprehensive $T_{\text{E}}X$ Archive Network (CTAN). *Class*. 2020. URL: <https://ctan.org/topic/class> (visited on 2021-08-21).
- [24] International Organization for Standardization. *ISO 8601*. 2019. URL: <https://www.iso.org/iso-8601-date-and-time-format.html> (visited on 2021-08-22).

- [25] Jobst Hoffmann. *listings – Typeset source code listings using L^AT_EX*. 2020-03-26. URL: <https://ctan.org/pkg/listings> (visited on 2021-08-23).
- [26] Will Robertson. *tocloft – Control table of contents, figures, etc.* 2020-01-11. URL: <https://ctan.org/pkg/tocloft> (visited on 2021-08-23).
- [27] Christian Feuersänger. *pgfplots – Create normal/logarithmic plots in two and three dimensions*. 2021-05-17. URL: <https://ctan.org/pkg/pgfplots> (visited on 2021-08-23).
- [28] The L^AT_EX Team. *hyperref – Extensive support for hypertext in L^AT_EX*. 2021-06-07. URL: <https://ctan.org/pkg/hyperref> (visited on 2021-08-23).

GLOSSARY

Argument	(<i>Programming</i>) The actual information stored in a variable.	5, 12, 16–21, 27–29, 31, 33, 45–47, 58–60, 69, 74, 106
Bookmark	(<i>LaTeX</i>) A marker used to identify a section, a figure, a table, and more inside a document. It comes before the description of said object or place, such as the numbered marker before a section title. It can be created with numbers, letters, or a combination of both.	19, 58, 59, 69, 70, 72
Citation	Similar to a reference, but specific to bibliographical sources.	5, 21, 49, 51, 69, 106
Class	(<i>LaTeX</i>) A kind of macro that defines a document layout standard.	iv, 3, 5, 15, 45, 46, 106
Command	(<i>LaTeX</i>) A kind of macro that reserves special words to define and alter a document's behavior.	iv, 3, 5, 7, 9–12, 15–23, 25, 26, 28–33, 35, 39, 40, 45, 47, 48, 51–53, 56, 58–60, 62–64, 71, 73–75, 105, 106
	(<i>Programming</i>) A code directive to perform a certain task.	4, 5, 39
Conditional	(<i>LaTeX</i>) A kind of macro that handles decisions.	v, xv, 63, 72, 106
Definition	(<i>LaTeX</i>) A kind of command that uses TeX primitives.	iii, 5, 45

Environment	(<i>LaTeX</i>) A kind of macro that defines a restricted scope where only specific behaviors are allowed.	xv, 3, 7, 12, 14, 19, 23–33, 41, 43, 47, 59, 60, 68, 70, 71, 74, 106
Group	(<i>LaTeX</i>) The restricted scope of an environment.	7, 12
Hook	(<i>LaTeX</i>) A kind of macro that defines special behaviors when intercepting code calls, events or messages.	xv, 106
Hyperlink	A link from a hypertext file or document to another location or file, typically activated by clicking on a highlighted word or image on the screen. Hyperlinks can be found in citations, URLs, general references, acronyms, etc.	v, 19, 20, 22, 23, 47, 51, 68, 69, 72, 107
Key	See parameter.	59, 74
Label	(<i>LaTeX</i>) The argument used by a referencing command to establish the link between an object and the place it's being called from.	19–22, 24, 31, 32, 69
Loop	(<i>LaTeX</i>) A kind of macro that handles repeated sequences of code until met a certain conditional statement is met.	xv, 106
Macro	(<i>LaTeX</i>) Any set of code that can be declared and called inside a document. It encapsulates classes, packages, environments, commands, conditionals, loops, hooks, variables, and more. It can be thought of as shorthand code that abbreviates a more complicated sequence of code.	ii–iv, xv, 3, 5, 7, 15, 21, 23, 25, 26, 28, 30–32, 40, 41, 43, 45, 47, 48, 51, 53, 56, 58, 59, 62, 63, 68, 69, 71–73, 75, 105–107

Package	(<i>LaTeX</i>) A kind of macro that defines stylistic attributes and enhancements to the capabilities of a document.	iii, v, 3–6, 19, 21, 23, 25–30, 32, 33, 39, 40, 45–47, 49, 51–53, 55, 57–60, 62, 63, 66, 68–71, 73, 74, 106
Parameter	(<i>Programming</i>) A kind of variable used as input to pass information to other parts of the code, across scopes.	7, 16–18, 46–49, 56, 59, 60, 69, 71, 75
Primitive	(<i>LaTeX</i>) A built-in, building-block macro of the TeX engine.	105
Reference	A specific mention that brings attention to a specific object in a document. Referencing in LaTeX is provided with hyperlinks.	ii, iii, 6, 19–24, 64, 69, 73, 105, 106
Source	A document or work that serves as the basis, proves, supports or supplements a piece of information.	iv, v, 5, 19, 21, 22, 43, 44, 49, 51, 93–100, 102, 105
Value	See argument.	17, 18, 21, 23, 30, 33, 35, 49, 59, 62, 70, 74
Variable	(<i>Programming</i>) Stored information with symbolic representation.	29, 35, 63, 105–107