The **Toptesi** package
Typesetting a PhD thesis with LaTeX

Claudio Beccari

Abstract

This tutorial uses the information given in the previous five ones in order to describe how to use the TOPtesi LaTeX package to typeset a PhD thesis. This package has a specific option to configure the typesetting of such a kind of thesis in the format agreed upon by ScuDo, the doctoral School of Politecnico di Torino.

Sommario

Questa lezione raccoglie l'informazione fornita dalle cinque precedenti al fine di descrivere l'uso del pacchetto TOPtesi per produrre con LaTeX una tesi dottorale. Questo pacchetto dispone di una specifica opzione per configurare la tipocomposizione della tesi nel formato concordato con la ScuDo, la Scuola di Dottorato del Politecnico di Torino.

1 Introduction

The TOPtesi LaTeX package (Beccari, 2019c,a)[1] to produce theses of different levels has been around for many years; the successive previous versions started to become too complicated and cumbersome. Now it has a modular structure and a suitable module is selected by expressing options in the form of *key* = *value*; in particular the type of doctoral thesis to be typeset at the Doctoral School of Politecnico di Torino is selected with the *key* = *value* pair set to tipotesi=scudo. The doctoral thesis style and structure are specific for ScuDo; several other options select other styles for other thesis types with other structures.

2 The class structure

Figure 1 shows the various modules that form the TOPtesi bundle. Each module is a file by itself and options are specified to select which module to use in order to typeset the desired title page and to configure the preamble of the document so that the thesis fulfils its requirements.

The user can specify his/her preferred packages; not any package, because it is important to avoid conflicts with the bundle modules and settings; but there is an ample choice.

Notice that the input source file `.tex` is fed to a toptesi.cls class, that receives the options and passes them to the selected modules. The first one is toptesi.sty; why another package when the class could do everything is desired? Simply because any user can choose to employ the `.sty` package with a different class, for example book.cls, or report.cls, or any other compatible class.

Myself, as the author of TOPtesi, I think it is not worthwhile to do such a mixture, but my opinion is evidently biased. I just want to emphasise that the possibility does exist, but unusual classes might form an unusual couple with toptesi.sty, and might perform in a bad way. I cannot tell which classes are compatible with toptesi.sty; I just can say that report.cls is the base class on which toptesi.sty works by default. Compatibility exists with book.cls, but I did not made any tests with other non standard classes.

Evidently the main class and the main package, with the possible support of the user loaded packages, and possibly with the further bundle topcoman.sty module, cooperate to the form of the output `.pdf` file typeset contents.

The various *values* assigned to the tipotesi key, select eight different modules to typeset eight different thesis types with different title page arrangements and internal structures.

The topfront.sty module is maintained for backwards compatibility; but this module is selected if no other module has been chosen (no option or no value specified) or when a wrong or misspelt *value* has been specified. Chances are that the compilation ends correctly, but if the user gives an attentive look to the result s/he might notice, for example, wrong hyphenation or different labels on the pieces of information that have been input, for example a high school name in place of a university name. But these occurrences should ring a bell to the user in order to inform him/her that some option value is wrong.

3 The option values

Let us describe the eight options

tipotesi=triennale Prepares a title page suited for a bachelor degree final work and a suitable structure; no extra packages are loaded besides the few ones that shall be described in

[1] The bibliography at the end of this paper contains many references to bundles and packages shipped with any TeX system complete installation, be it TeX Live or MiKTeX. The references that carry the notice "Readable with. . . " are all already in the users' computer where the TeX system is installed so no Web search is necessary.

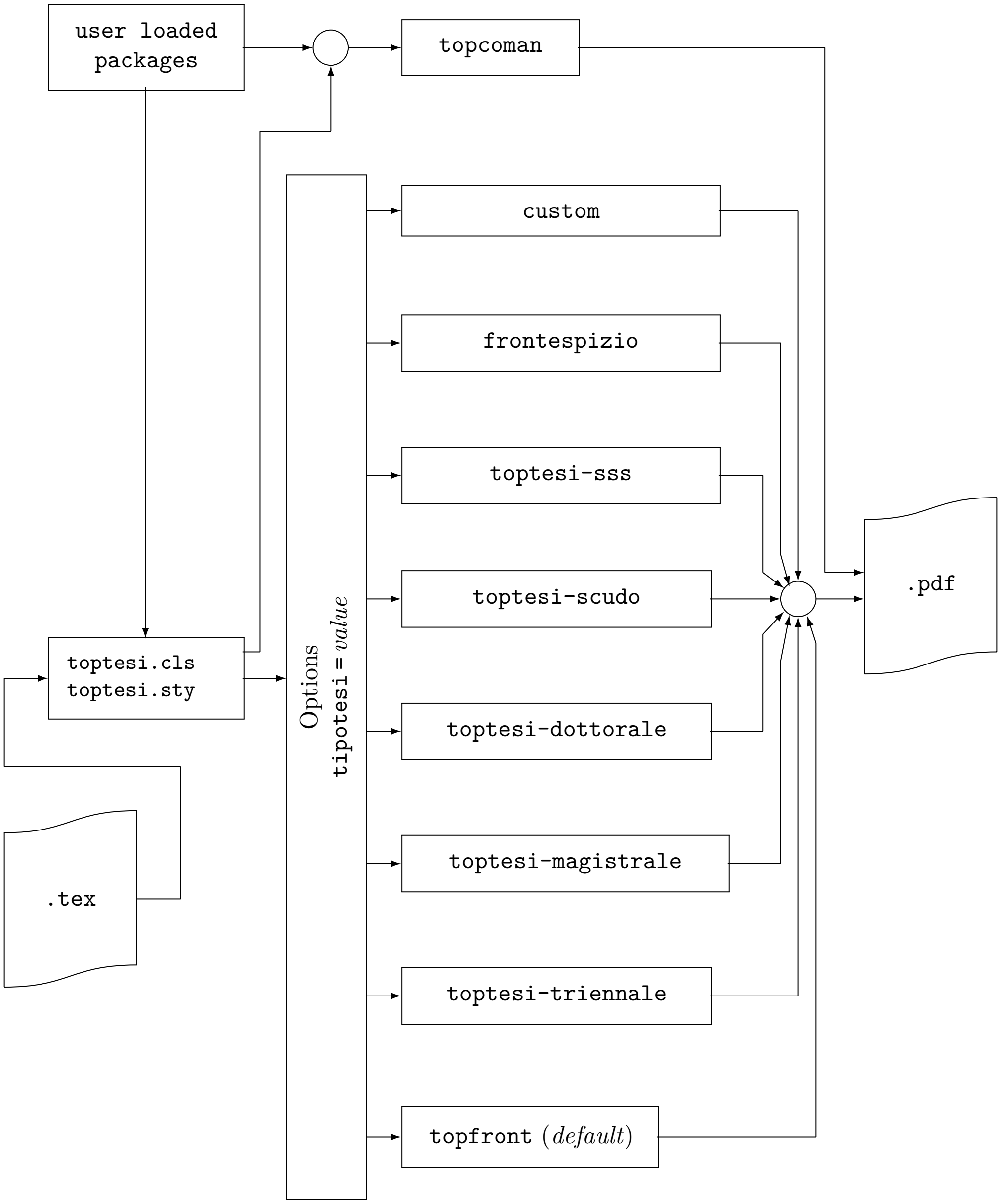


FIGURE 1: Version 6.x TOPtesi bundle flow diagram

the following sections. The default language is Italian, but with the use of the declarations `\english` or `\italiano` (or with suitable groups or environments) it is possible to switch back and forth between these two languages.

tipotesi=magistrale Prepares a title page suited for a master thesis and a suitable structure; no extra packages are loaded, as with the bachelor degree option and, similarly, it is possible to switch back and forth between Italian and English.

tipotesi=dottorale Prepares a title page and a suitable structure valid for doctoral theses in general. This option should be suitable for PhD students in other doctoral Schools. Language switching is similar to the previous thesis types. No extra packages are loaded.

tipotesi=scudo Prepares a title page and an internal structure suitable with dissertations to be defended after frequenting the Doctoral School of Politecnico di Torino. Some other packages are preloaded, and the *default language is English*.

tipotesi=sss This unusual and unique option is used to typeset a high school final work. This module was asked for by high school students who were already familiar with LaTeX. Unfortunately the Italian Ministry of Education and the Parliament decided to abolish the final work report and this year 2019 is the first year where the final high school state exam does not require the preparation of a such report. The module is conserved for backwards compatibility.

tipotesi=frontepsizio This option excludes the internal module topfront to be used to typeset the title page, but it loads the frontespizio package that is alternative to this bundle module since they are mutually incompatible. This option does not forbid the user to employ the alternative package, but the option must be specified in order to avoid loading what might be in conflict with the chosen package.

tipotesi=custom This option leaves the user completely free to compose the title page "by hand", that is the user can use a titlepage environment, where s/he can put whatever piece of information, typesetting it with any font of any size, any family, any series and any shape available for that font. The structure of the document remains the default one and almost any extension package chosen by the user can be loaded.

tipotesi= No option or no value or a wrong value is specified; this is maintained for backwards compatibility and as a fall back style in case of errors.

This variety of choices requires a thick documentation; the file `toptesi.pdf` and `toptesi-it.pdf` are part of the bundle; they are directly accessible through the `texdoc` terminal command with the following syntax[2]:

```
texdoc toptesi.pdf
texdoc toptesi-it
```

The first text is completely in English, but most of it is the documentation of the code; in spite of this, when the codes of the various modules are commented, there are many explanations, suggestions and examples of how certain commands may be used, or certain solutions may be tweaked, or certain strings may be adjusted to the users' needs.

The second text is mostly in Italian, but the parts that are connected to the tipotesi=scudo option are in English; it deals mostly on how to compose the various thesis types and it explains why there are so many differences in the title pages.

4 General information for various thesis types

The structure of a thesis may vary according to its level and to the scientific domain it is about. Let us deal only with the three university levels: bachelor, master, and doctoral thesis types.

4.1 Bachelor final work or bachelor thesis

In some countries and in some universities the bachelor university courses are completed with a final work that generally deals with what has been learned during the degree course: it is "just" an application of what the candidate did actually learn during his/her studies. No one expects that the final work contains new theoretical aspects. The value of the final work is measured through the quality of the application. Sometimes this final work is labelled as bachelor thesis; actually it is not excluded that something new appears in such a thesis, but in general it is a smart application of known practices.

This has an influence on the structure of the thesis; in the sense that initial chapter(s) on the state of the art are usually missing, and the same is valid for the final chapters: conclusions, further investigation, possible developments, and the like.

Therefore this type of document contains the title page, seldom it contains a legal page, certainly it contains the table of contents, and possibly the list of figures and that of tables. There will be an introductory chapter that describes what the whole thesis is about and what was its purpose; one or more chapters describing what has been done; such chapters, depending on the scientific domain, might be rich of figures, drawings, photographs, and the like; they will contain tables of measures, or material specifications, or tables of temporary

[2] In the first case the extension is compulsory, because without it the second file is opened.

or final results, or similar tables; if the thesis deals with the design of an equipment, of a machinery, of a plant, or a building, the executive drawings may require large sheets of paper, folded in a particular standard way, or outside the thesis collected in cardboard tubes.

Some of this material may be created with LATEX, but some, especially large drawings, require special drawing equipment and/or special software and special printers. Here I will not deal with the special resources external to the typesetting procedures.

Text and tables are done in LATEX; the latter are more difficult to typeset, because it is difficult to preview what is actually desired to obtain.

4.2 Master thesis

Master theses are real theses. They are supposed to be defended in a formal examination session where the candidate should describe his/her work and defend it against the objections of the examining committee. This type of examination is the traditional one that ends with a laurel wreath (*corona laurea* in Latin) to be worn by the new Master. Several mottos contain the word, in particular that of the late Institution of Electrical Engineers (IEE, now merged with the IET): «Lauream ferat qui meruit». The academic title in Latin is "Magister", the word where the name "master" comes from.

A master thesis should contain something really new, some new theory, or some demonstration that some conjecture is true. Sometimes a very good master thesis may receive the declaration of being worth of publishing.

Evidently the introductory part and the concluding part of a master thesis are wider and more detailed than the corresponding parts of a bachelor thesis; nevertheless it is not the number of pages the element that distinguishes a really good thesis; among the glories of Politecnico di Torino there is an engineer who stated and proved the theorem of virtual works, now known as the Theorem of Castigliano; Castigliano's thesis did not go over thirty pages.

The typographical elements of a master thesis are more or less the same as those of a bachelor thesis. The title page is different and the legal page is often required.

4.3 The doctoral dissertation or PhD thesis

Similarly, the doctoral dissertation is an even more complete work of research than the master thesis; very often it reports on a research that the candidate developed during his/her study period on a PhD degree course, and partial results have already been published on scientific journals.

The Latin academic title is "Doctor", where the modern title comes from, including the adjective "doctoral" that qualifies the thesis, often called dissertation.

The typographical elements of a doctoral dissertation are not very different from those already described for the other theses. The title page might contain a lot of information concerning the formal examining commission.

The disciplines dealt by a doctoral dissertation are the most varied and go deep into the details. Every scientific discipline has its jargon and its habits. It is possible to distinguish a thesis in hard disciplines from those in soft ones by looking at the footnotes and bibliographies; the body of the text may contain a lot of mathematics, or a lot of verbatim citations of other authors. Here comes the utility of a specific LATEX class in order to properly typeset these elements.

4.4 The thesis bibliography

All thesis types contain a bibliography; single references in this bibliography are cited again and again in the text body.

There are many ways to typeset bibliographies and to cite their references. LATEX offers several packages and programs to handle these typographical units.

The best way to handle a bibliography is to write a bibliographic database where each record describes a reference in a way suited to its nature: a book, a report, a chapter, a contribution to a collective book, and article in the proceedings of a conference, an article in a journal, another thesis, and so on; each type of reference may have different elements: title, author(s), publishing house with its location, volumes of a collection of proceedings, dates, conference name and location, and so on. Such a database is precious; but is not in a form that LATEX can directly handle in order to produce the references in a recognised style and the citations in the form needed by a specific discipline. LATEX provides several bibliographic style files and the TEX system installation has at least two main programs that extract the necessary information from the bibliographic database and feed such information to LATEX in a form that allows it to typeset the bibliography with the proper style.

5 Lists and tables

Lists are of three main groups: itemising, enumerating, descriptive lists. The items listed into an itemising list are marked with a not alphabetic symbol that is equal for all items; more often than not this symbol is a bullet.

Enumerations number their items with ordered symbols, typically arabic numbers, upper- or lowercase roman numbers, lower- or uppercase letters of some alphabet. It is generally possible to label one or more items of an enumeration so as to recall them for reference purposes.

Descriptions are similar to itemisations, where each item is introduced by a word or a short phrase that is being described in the item body; but, differently from itemisations, each item may consist of one or more full paragraphs.

In technical writings itemisations are part of the paragraph that introduces them; therefore items start with a lower case initial first word and never end with a full stop, except possibly the last item in the list. This implies that the list items cannot contain more than a fraction of a paragraph; rather it contains simple sentences, or simple phrases, or single words; punctuation at the end of each item may be made with commas and semicolons, never periods or other terminal punctuation signs; the item final punctuation may be omitted in a displayed itemisation.

On the opposite, enumeration and description items form paragraphs and start with an uppercase letter and end with a full stop; such items may contain more than a single paragraph.

These details must be taken care of by the user, they cannot be handled in an automatic way by LATEX.

Tables may be of different types; they may be small or large, remain within the margins of the printed block, or be wider; they may be in a normal position or rotated 90° counterclockwise; they may occupy one page or may be several pages long.

For each of these tables LATEX has various solutions and it should be the user responsibility to know what to do to handle a specific table and/or which package to load that has a ready to use solution.

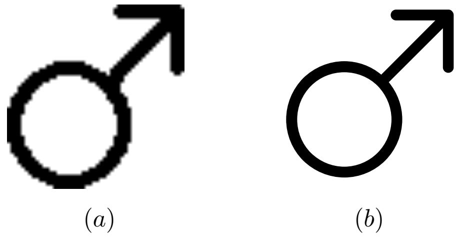
In all cases the user should plan a table very carefully; yes *plan*, because a table is not simple text, where LATEX can do a beautiful job for splitting a sequence of words or similar entities into lines of equal measure while minimising the inter word space. Tables are bidimensional objects and must be handled in a different way; LATEX can do many things except typesetting in a beautiful way a badly planned table.

6 Images of any kind

LATEX, as a mark up language, has all the necessary commands for both drawing certain images and importing certain types of already made images. In any case it is impossible to produce any image in a LATEX document simply by dragging a graphic file from a folder to the editing window, as it is possible to do, in certain cases, with some word processors. Let us separate these two operations.

Importing images This is relatively simple because the fundamental command `\includegraphics` is self explanatory. The point is how to use correctly the various options accepted by this command, and which kind of images the command can include; for what concerns the options, they will be discussed further on; for what concerns the images, all typesetting engines, except the simple `latex`, can import images in the formats (*a*) EPS, PDF, and MPS, that are or may be of vectorial nature; and (*b*) JPG and PNG, that refer to bitmapped formats; such image files may be imported if and only if they are well formed and contain the metadata that specify their characteristics, in particular their natural dimensions; otherwise it is necessary to include such dimensions within the options to the `\includegraphics` command. This method can be used also for other bitmapped image formats, but it is often difficult to discover the correct image natural dimensions. In any case such bitmapped images cannot be scaled at will: if they get downscaled they might loose definition; while if they are upscaled, their granularity becomes very evident: see figure 2 to notice the difference with a bitmapped character and a vectorial counterpart, both enlarged by a very large scaling factor.

Drawing images Any TEX system installation contains several packages to make drawings; the native LATEX environment picture, that can be extended by loading package pict2e ((Gässlein *et al.*, 2016); this extension was already documented by Leslie Lamport in the second edition of his LATEX manual, (Lamport, 1994)); a larger extension is obtained by loading package curve2e, (Beccari, 2019b), that extends pict2e — figure 1 was drawn within a picture environment extended with package curve2e. Beautiful drawings may be created with the powerful packages TikZ (Tantau, 2019) and pgfplots (Feuersänger, 2018); or with the even more powerful package PSTricks; even with METAPOST, (Hobby, 2018); with the interactive interface `asymptote` (Hammmerlindl *et al.*, 2018); with the external program `gnuplot` (Miklavec, 2013).
Furthermore it is possible to download from the Internet other programs that are LATEX aware. The cited *packages* allow to execute "programmed drawings" while the cited *programs* generally work in an interactive mode so that the user works with his/her mouse instead of writing code. The subject is too large to be described here even if some information has been given in the previous tutorials. For what concerns the packages and the programs included in any complete TEX system installation, the documentation may be read by simply typing in a command window `texdoc` followed by the name of the



(*a*) (*b*)

FIGURE 2: Comparison between an enlarged bitmapped charcter and its vectorial equivalent

> package or of the program; this is what has been recalled in the bibliography of this paper.

It is evident that if vectorial drawings are available their rendering is much better than with the bitmapped ones. For photos the lossy compressed bitmapped JPG format is acceptable, provided that the pixel density (pixels per inch) is not too small: a density of 150 pixels per inch is generally good, but remember that the graphic file quadruples its size if the pixel density is doubled; the user must seek a compromise between pixel density and quality of the rendering. The lossless compressed bitmapped PNG format is better for line drawings compared to the JPG one, but the compression is less effective than with JPG drawings; at the same time the lossy nature of the JPG format may degrade line art by superimposing displaced phantom replicas of the main drawing.

Vectorial formats are preferable, but beware: a vectorial file, such as EPS or PDF, may contain bitmapped images; therefore control very attentively before using such a "fake" vectorial image; such control is very simple: open the vectorial image to be tested and enlarge it at least by a factor of 10: if the quality of the image does not change it is truly vectorial, otherwise it is a vectorial container of a bitmapped image.

7 Mathematics

The mathematics of hard sciences (experimental sciences) deal with *quantities*, i.e. symbols that represent the pair *measure plus unit of measure*, rather than mathematical variables; it is fundamental what has been described in a previous tutorial on typesetting mathematics and on using package siunitx (SMITH, 2018).

Nevertheless there are some international regulations, (ISO-31/XI, 1978), that require a specific usage of math fonts for physics and technology; in general mathematics are typeset with special characters that are classified with the name of *math groups*; these include the operators, letters, symbols, and extensible delimiters groups; special documents may require other groups. The latter two groups are self explanatory; the former ones may be confused with the roman and the italic fonts; they are not to be confused, because even if some of the glyphs appear identical, their properties are different. With the letters group the difference is very noticeable: the phrase *different affinity* (written with a text italic font) becomes $differentaffinity$ with the letters group, and it becomes *differentaffinity* with italic font used by the `\mathit` command. The lack of ligatures and inter-word spaces is evident with the letters group; with `\mathit` the ligatures are still there, but the inter-word spaces are still missing.

The ISO regulations (ISO-31/XI, 1978) in general require specific usage of bold face series; roman, italic, sans serif upright, and sans serif oblique have special meanings, the details of which may be found in the mentioned regulations and in other texts; I would suggest the freely downloadable manual (THOMPSON and TAYLOR, 2008), that explains how to use units of measure and how to write mathematics; it also adds several pieces of information that are very difficult to find elsewhere.

8 Nomenclature and glossaries

Strictly speaking a nomenclature list or a glossary are not needed in a thesis of any type. But in doctoral theses, that deal with research, a nomenclature, or glossary, or acronyms list may be useful.

LaTeX offers several packages to typeset such lists; the most common packages are the nomencl (VEYTSMAN *et al.*, 2019), glossaries (TALBOT, 2019), and acronym ones (OETLIKER, 2015). It should be stressed that such packages may compose several such lists, but each one is specialised in one of them; they require the use of an external program to sort and format each entry; more is said below in connection with the ScuDo doctoral theses.

9 Indices

Again indices are not necessary in any thesis type. Nevertheless the doctoral theses, due to their advanced contents, may benefit from the presence of one or more indices.

Here we suggest the imakeidx package by GREGORIO (2016) because it can solve all the problems of index production; it is highly configurable so that it is possible to produce indices with a specific name different from the default one and from the names of other indices in the same document; it is possible to decide to compose in one, two, or three columns, and a specific index style for the entries, plus a nice series of other customisations. The tipotesi=scudo option preloads this package.

The documentation of imakeidx (GREGORIO, 2016) explains not only the usage, but offers a variety of examples and various tricks to tweak the entry style.

10 Archivable format

Most universities require every student to submit a file containing his/her thesis to be archived for legal documentation and possible information for other people; Politecnico di Torino is one of them.

Any archivable document must fulfil the requirements stated by specific ISO regulations. Such regulations have been published and updated several times, (ISO 19005-1, 2005; ISO 19005-2, 2011; ISO 19005-3, 2012); they are labeled by acronyms such as PDF/A-1a, PDF/A-1b, and so on.

All regulations require the PDF format; they require also certain requirements on imported pictures and their color profiles; they require that the used fonts are embedded in the file, at least the subsets of the used characters, and that the fonts be encoded according to the UNICODE standard; no character in any font should have a vanishing width. Moreover some of these regulations require that the contents of the file be of the kind known as *Tagged PDF*.

As of today, the TeX system typesetting programs based on the LaTeX mark up cannot produce tagged PDF files; in this very GuIT Meeting another speaker describes the work that is being done in order to introduce this feature into the output PDF files. Possibly in the near future all the programs that use the LaTeX mark up, at least LuaLaTeX, will be capable of producing tagged PDF files; LuaLaTeX is also convenient in order to handle UNICODE encoded fonts.

pdfLaTeX cannot be used because it can handle only one-byte encoded fonts; the first “page” of the UNICODE standard contains only the ASCII one-byte encoded subset, but this subset does not contain either accented characters, or accent glyphs to superimpose to the unaccented ones; therefore pdfLaTeX is out of the game.

XeLaTeX, also, is out of the game; it is not impossible to use it, but in order to end with an archivable format it is necessary to do some post-processing and to accept some compromises.

Therefore only LuaLaTeX is effectively suitable for producing PDF/A files, at the moment just non tagged PDF ones. Therefore the ISO standard obtainable is the one labeled PDF/A-1b.

Some universities would like to have archivable documents accessible to impaired readers; the official ISO regulations for this task are still under discussion, but work is being done also in the TeX world, in particular by the Team on Accessibility at the University of Turin. This Team has already produced a package suitable for use by blind or impaired vision readers (Ahmetovic *et al.*, 2018); work is still in progress.

11 Comments

The preceding sections mostly describe the typesetting problems that are common to any thesis type. Now it is time to concentrate on PhD theses, in particular those that are conformant with the requirements of the ScuDo Doctoral School of Politecnico di Torino.

12 TOPtesi and its tipotesi=scudo option

What has been described in section 4 about thesis types in general applies also to those to be developed at the ScuDo Doctoral School unless something special is unique for the latter ones.

The specific option tipotesi=scudo, to be specified to the toptesi class, configures the title page, the copyright or legal page in the proper way, and preloads a number of packages that are considered essential for doctoral theses in the scientific domains that form the various PhD degree courses in the ScuDo Doctoral School.

All or most of these degree courses find their counterparts in the international Institutions that deal with engineering professions, including some professions dealing with architecture. I mention only the Institution of Electrical and Electronics Engineers (IEEE) simply because it is closer to my scientific interests, but this is not a selective choice.

12.1 Facility for using either pdfLaTeX or LuaLaTeX

With the tipotesi=scudo option the preamble of the main source file is suitable to detect which typesetting program is being used: pdfLaTeX or LuaLaTeX. The former is perhaps the most suitable to use when working on the thesis drafts before the final version; the latter is suitable for producing the final PDF/A compliant PDF file.

Actually it is up to the user to configure the preamble for this task: the TOPtesi bundle contains a `toptesi-scudo-example.zip` file that has a complete commented example that is worth studying in detail. Neglecting the comment lines, this example file preamble contains the following lines:

```
\documentclass[%
  corpo=12pt, % optional;
              % default font size:= 10pt
  twoside, % recommended
  tipotesi=scudo,
  mybibliostyle, % necessary only if
                 % bibliography is typeset
                 % with different style
  numerazioneromana,% roman page numbering
                 % just for testing
                 % don’t use in your thesis
]{toptesi}

```

```
14 \ifPDFTeX
15   \usepackage[utf8]{inputenc}%
16   \usepackage[T1]{fontenc}
17 \fi
18 ...
19 \ifPDFTeX % using pdflatex
20   \usepackage{lmodern} % Default
21   %\usepackage{newtxtext,newtxmath}%
22       % Times eXtended for text and math
23   %\usepackage{fourier}% Utopia,
24   %    Helvetica and "monospace = ?"
25 \else % using lualatex (or xelatex)
26   \usepackage{fontspec}
27   \defaultfontfeatures{Ligatures=TeX}
28   \setmainfont{Libertinus Serif}
29   \setsansfont{Libertinus Sans}
30   \setmonofont{Libertinus Mono}%
31               [Scale=MatchLowercase]
32   \usepackage{unicode-math}% add special
33            % math style option here
34            % for example [math-style=ISO]
35 % define one math font
36    \setmathfont{Libertinus Math}%
37 \fi
38 ...
39 \makeindex[intoc]% collect material
40                 % to one index and list
41                 % the index in the
42                 % table of contents
43 ...
44 \ifmybibstyle % customise bibliography
45   \usepackage[autostyle]{csquotes}%
46                % necessary for biblatex
47   \usepackage[backend=biber,
48               style=philosophy-classic,
49               scauthors=all,
50               sorting=nyt,
51               natbib]{biblatex} % LaTeX
52               % specific bibliography
53               % handler
54   \addbibresource{\jobname.bib}%
55            % bibliographic data base(s)
56 \fi
57 ...
58
59 \ifPDFTeX \usepackage{indentfirst}\fi
60 \raggedbottom
61
62 \begin{document}
```

This (partial) preamble needs some comments.

1. The lines that contain only three dots, denote skipped material that is not worth commenting here; they mostly deal with variants required when the PDF/A version of the thesis is required.
2. Lines from 1 to 11 show the document class statement with the options; please notice that the last two options are discouraged; option numerazioneromana is for using roman numerals in the front matter; actually this style of numbering is not necessary in modern documents; it was necessary when typographies worked with metal type. In some disciplines this traditional numbering is still appreciated, but there is absolutely no need to use it simply because it is available; use it to test the result, but avoid using it in your real thesis.
3. The mybibliography option is discouraged because in scientific publications dealing with the disciplines connected with engineering, the concise numerical style is highly preferred. Nevertheless some disciplines (or some PhD students) prefer the author-year style or some other style, so they can implement their choice, but they have to explicitly use a code similar to the one shown in lines from 43 to 55.
4. The various tests produced by the `\ifPDFTeX` conditional command detect if the thesis is being typeset by using pdfLATEX; else LuaTEX or XƎLATEX is assumed, even if XƎLATEX is discouraged as explained before.
5. In lines from 13 to 16 the necessary encodings are specified if pdfLATEX is being used; it is not necessary to specify such encodings with LuaLATEX because UNICODE (or UTF-8, UNICODE Transformation Format) is assumed for both input and output.
6. Lines from 18 to 24 provide the normal use of the Latin Modern Fonts but offer a couple of (commented out) alternatives that may replace the standard Latin Modern fonts: the Times fonts or the Utopia ones. According to my experience, the Latin Modern fonts may appear "too common", but there is a reason; they have optical sizes and are much better suited for technical documents than any other font that does not exhibit this feature. But, of course, this is a question of personal taste.
7. Lines from 26 to 36 configure the necessary fonts for typesetting with LuaLATEX; the font handler fontspec (Robertson, 2019b) is loaded; by default it uses the OpenType version of the Latin Modern fonts; in this example the Libertinus serif, sans serif, and teletype text fonts are loaded; since math is an important part of any technical discipline, the unicode-math package (Robertson, 2019a) is loaded and the Libertinus Math font is selected. A comment specifies that it is possible to specify the math-style=ISO option to the unicode-math package. I strongly recommend to use this option because it fulfils almost all the requirements of the ISO regulation concerning the style of writing prescribed for physics and technology.
8. Line 39 configures the only index that is being produced; the command not only enables collecting the material for the index, but also specifies that an entry for such an index should appear in the table of contents.

9. From line 44 to line 56 there are the conditional settings to configure a customised bibliography; this is just a model, but by experimenting with the example `toptesi-scudo-example.tex` the user can see what it typesets by commenting or uncommenting the class option line 6. Of course every time a style change is done it is necessary to run the typesetter, then the bibliography processor `biber`, than again the typesetter; certainly the user has the necessary experience on these points and it should be superfluous to recommend this procedure.
10. Eventually in line 59, if pdfLATEX is being used, the small package indentfirst (Carlisle, 1995) is called for indenting the first paragraph of each sectional unit; by default such first paragraphs are not indented, while with LuaLATEX they are indented. This little package provides an identical behaviour with both typesetting programs.
11. The last declaration in line 60 specifies that pages should be typeset with a ragged bottom. Of course this is an optional setting; this particular example file contains many large objects, therefore, in order to vertically justify the text block, some glue would be inserted between paragraphs and above and below objects in display; this procedure guaranties that all text blocks have the same height, but that filling white space is very annoying. Personally I prefer a few pages to have uneven bottoms than to have filling white space. Of course, with a different contents, the specification `\raggedbottom` might be superfluous.

The user can examine the source file of `toptesi-scudo-example` and examine what has been omitted from this description. Also the typeset document explains certain details, besides displaying several examples.

The user can copy the `.tex` file to another file while changing its name: and s/he can play experiments by changing some settings, or by testing some other functionalities.

12.2 Splitting the source `.tex` file

The use of commands `\includeonly` and `\include` help maintaining everything connected with single chapters or initial or ending parts of the thesis pretty clear and separate. The preamble of the example file contains this short stretch of code:

```
\includeonly{%
Chapter1/chapter1,%
Chapter2/chapter2,%
Chapter3/chapter3,%
Appendix1/appendix1,%
Appendix2/appendix2,%
References/biblio%
}
```

The various chapters are simple files starting with the `\chapter` command and possibly ending with `\endinput`; they do not contain any preamble and are saved in their own folder; all folders must be subfolders of the one that contains the main file.

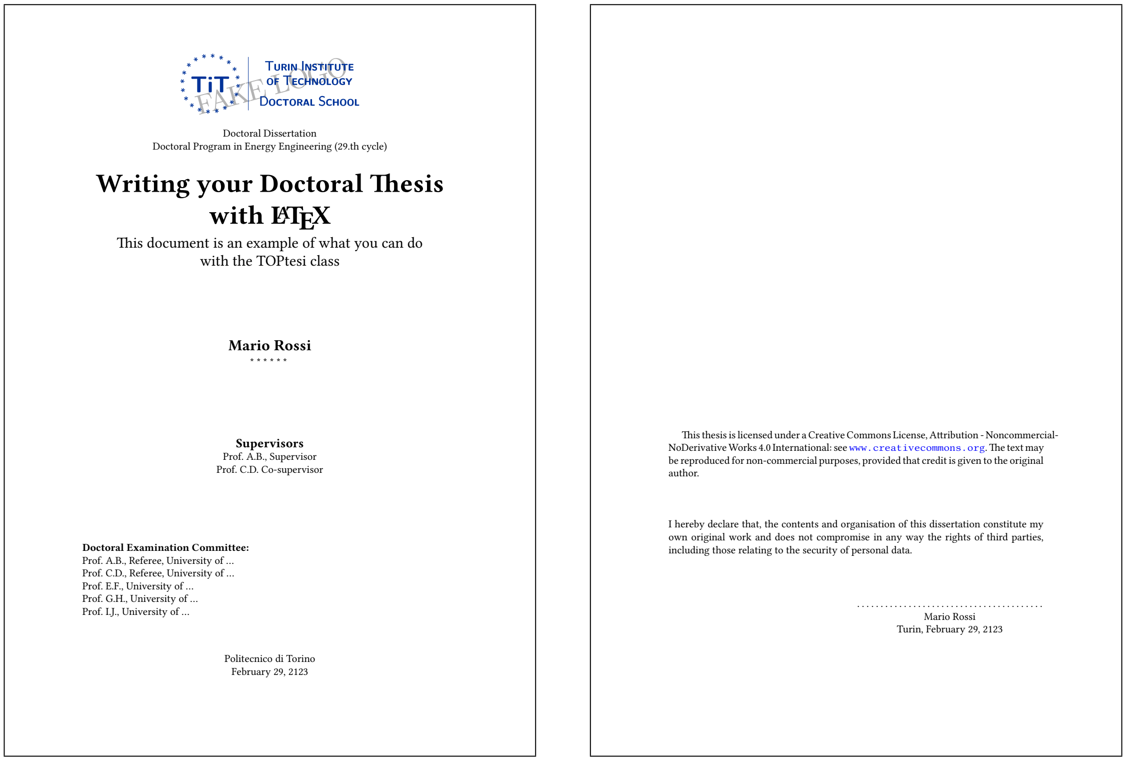
The list of files to be included are written one per line; in this way if one comments all lines except one, it is possible to typeset just that file, with the advantage that all cross reference data of the other files remain available, of course only if the other files have already been typeset. In this way one might compile the first chapter alone; when it is almost complete, its line gets commented out and the next line is uncommented; therefore a new typesetting run is done, and only chapter 2 is typeset and any cross reference to chapter 1 is correctly used. The process proceeds quickly as far as the end of the document, without loosing time in typesetting again and again chapters that have already almost reached their final state. When all chapters are ready, it is time to uncomment all the included file names, in order to fix some residual details, and the final typesetting run creates the whole document.

Of course this trick is applicable to any file to be typeset with any LATEX typesetting engine; here it is recommended because the completion of a doctoral thesis, with its delicate structures and advanced technical language in English (which is the default language for the ScuDo doctoral theses) requires a very attentive reading and correcting of the various drafts; splitting the job into simpler parts eases the task.

12.3 The ScuDo title page and the legal page

Again the example file shows how to typeset the title page and the legal page. The code of the file is highly commented in order to explain variant possibilities. Here I strip the comments out and describe the basic ThesisTitlePage contents.

```
\begin{ThesisTitlePage}
\PhDschoolLogo{TiTDocScCropped}%
           % Fake logo for this example
\ProgramName{Energy Enginering}
\CycleNumber{29.th}
\author{Mario Rossi}
\title{Writing your Doctoral~Thesis\\
  with \LaTeX}
\subtitle{This document is an example
   of what you can do\\with~the~TOPtesi
   class}
\SupervisorNumber{2}
\SupervisorList{%
   Prof.~A.B., Supervisor\\
   Prof.~C.D. Co-supervisor}
\ExaminerList{%
```



Doctoral Dissertation
Doctoral Program in Energy Engineering (29.th cycle)

Writing your Doctoral Thesis with LATEX

This document is an example of what you can do with the TOPtesi class

Mario Rossi
* * * * * *

Supervisors
Prof. A.B., Supervisor
Prof. C.D. Co-supervisor

Doctoral Examination Committee:
Prof. A.B., Referee, University of …
Prof. C.D., Referee, University of …
Prof. E.F., University of …
Prof. G.H., University of …
Prof. I.J., University of …

Politecnico di Torino
February 29, 2123

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial-NoDerivative Works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . .
Mario Rossi
Turin, February 29, 2123

Figure 3: The title page and the legal page

```
Prof.~A.B., Referee, University of \dots\\
Prof.~C.D., Referee, University of \dots\\
Prof.~E.F., University of \dots\\
Prof.~G.H., University of \dots\\
Prof.~I.J., University of \dots}
\ExaminationDate{February 29, 2123}
\Disclaimer{%
\noindent I hereby declare that, the
contents and organisation of this
dissertation constitute my own original
work and does not compromise in any way
the rights of third parties, including
those relating to the security of
personal data.
}
\end{ThesisTitlePage}
```

The example makes reference to the PhD School logo; in this case the imported logo is a fake one, because the original ones are downloadable only by the students of the ScuDo doctoral school from a reserved Internet address that I cannot access; the School Registrar enables each student to access the site.

All the personal names of this example are fake ones; even Mario Rossi is the Italian equivalent of the proverbial John Smith in English speaking countries.

As it may be seen in figure 3 the logo is at the top of the page and the other pieces of information are distributed on the page with adequate labels. Such labels might be changed (when allowed), by using special commands that can be found in the English part of the already mentioned documentation file `toptesi-it.pdf`.

In figure 3 there are the statements for the licence related to the contents of the document, and the disclaimer where the signer, who is supposed to sign in original on the dotted line, declares that s/he respected both the intellectual property of others and that s/he did not violate the privacy rights of third persons. This disclaimer has been agreed upon with the Director of the ScuDo doctoral school and cannot be changed, while the “signature” part may be customised.

13 Structuring the thesis

It is not mandatory to segment the whole document in partial files, as already described, but is very useful.

It is very important to divide the inner material in front, main and back matters; each of these divisions may be further divided.

In table 1 it is possible to see the sequence of the major sections, front, main and back matter, and the placement of numbered and unnumbered chapter level segments of a PhD thesis.

Most of the major sectioning with TOPtesi is automatic and does not require intervention from the user. Possibly the user has to avoid using the starred or unstarred `\chapter` command in the front matter, because the first occurrence of this

Table 1: Stucture elements of a PHD thesis

Front matter	
Half title	Generally absent in a PhD thesis.
Title page	As specified by the ScuDo doctoral School.
Legal page	As specified by the ScuDo doctoral School.
Dedication	It is better to avoid dedications in PhD theses.
Acknowledgements	Very seldom necessary. See below
Foreword	Not necessary in a PhD thesis.
Table of contents	It may be followed by the list of figures and the list of tables.
Introduction	A PhD thesis should have an introduction, an unnumbered chapter that may appear in the table of contents. The author describes the structure of the thesis and shortly describes the motivation of performing his/her research in that particular domain; sometimes, and in very special circumstances s/he may acknowledge the support received by other people external to Politecnico di Torino, or external Institutions.
Main matter	
Numbered chapters	A sequence of structured chapters, divided in sections, subsections and other similar hierarchical divisions; generally sections below subsections are not numbered; even if they are, they do not appear in the table of contents.
Numbered appendices	Appendices, if any, appear at the end of the main matter only if they are more than one, and therefore are numbered.
Back matter	
Single appendix	A not numbered single appendix appears at the beginning of the back matter.
Bibliography	One or more bibliographies appear after the possible single appendix; generally there is just one bibliography, but sometimes the references may be arranged in separate lists. Often the search of information useful for the research at the base of the thesis was found in the Web. Each website entry should contain the date of the last visit, otherwise the information is useless.
Glossary	Glossaries, nomenclature lists, acronym lists, and the like are optional.
Index	Optional in a PhD thesis, but not useless; one or more indices are generally present in reference manuals rather than in theses.

command switches from front to main matter settings. Alternative commands for a summary or a dedication or an acknowledgements section are already available in order to avoid an unwanted shift to main matter.

But the problem may arise only with the Introduction; if this consists of a few pages and is not structured, an unnumbered chapter is in order, but if this introduction is long and possibly structured, then it is better that it is a numbered chapter and therefore the normal `\chapter` command is in order.

If a long not numbered and not structured Introduction is necessary the workaround is this:

1. Do not use roman numerals for the folios of the front matter.
2. Use the following code

   ```
   \setcounter{secnumdepth}{-1}
   \chapter{Introduction}
   ⟨Introduction body⟩
   \setcounter{secnumdepth}{2}
   ```

   In facts the level for numbering chapters (see table 2) is lowered to -1; then after the Introduction is finished and before using again the `\chapter` command the level is restored to its original value of 2. It is impossible to use groups to limit the scope of the new setting of the LaTeX counter to the group, because the `\setcounter` command assigns a value to the named counter in a global way.

14 The main matter

The main matter has all sectional units well numbered in a hierarchical way until a level that by default is 2; see table 2; they are listed in the table of contents to level 1. These values may be modified by changing the values contained in a couple of LaTeX counters:

```
\setcounter{secnumdepth}{⟨level⟩}
\setcounter{tocdepth}{⟨level⟩}
```

If the user wants to customise the table of contents in such a way that it contains also the entries for other sectioning commands, s/he can use the command `\setcounter` because LaTeX counters may be assigned values only this way; but remem-

Table 2: Levels of sectional units

Sectional unit	Level
part	-1
chapter	0
section	1
subsection	2
subsubsection	3
paragraph	4
subparagraph	5

ber: the value assignment is global and there is no grouping that may delimit its scope.

Obviously each chapter is long enough to be sectioned in a hierarchical way. Remember: it is meaningless to create a chapter with just one section, as well as to create one subsection within one section, and so on. Once a sectioning level contains any number of sectioning units of higher level, it cannot stop until a new section of the same level is started. More clearly: section 2 of chapter 3 is divided into subsections the first of which needs not be immediately after the title of section 2, but the last subsection of section 2 ends only when section 3 starts.

Any sectional unit contains plain text and may contain some material in display. Let us distinguish floating objects from fixed position objects in display. The former objects float until the typographical rules encoded into the class let them be extracted from their stack and actually inserted in the output pages. The latter are relatively large objects of different nature: they may be titles, formulas, tables, figures, and so on, and they cannot be broken across pages.

14.1 Formulas in display

A simple formula in display occupies a vertical space on the page that is equivalent to at least three lines of text; if the formula contains large operators and fractions it may occupy even more vertical space; if the formula is very long, it may occupy a lot of vertical space that cannot be broken across pages; arrays of formulas occasionally might be broken across pages if suitable commands are used and, certainly, if the array of formulas is not grouped by a large brace.

Let us make an example: everybody knows that the second degree equation[3]

$$x^2 + 2ax + b = 0$$

has its solutions given by

$$x_{1,2} = \begin{cases} -a \pm \sqrt{a^2 - b} & \text{if} \quad a^2 > b \\ -a & \text{if} \quad a^2 = b \\ -a \pm \mathrm{i}\sqrt{b - a^2} & \text{if} \quad a^2 < b \end{cases}$$

It is evident that the three forms of the solution cannot be split across page breaks (nor across columns when typesetting in twocolumn mode).

In any case the toptesi.cls class with option tipotesi=scudo preloads a number of packages; among these it preloads the amsmath (American Mathematical Society, 2018), amsthm (American Mathematical Society, 2017), and, only when typesetting with pdfLaTeX, amssymb (American Mathematical Society, 2013); therefore the whole machinery made available by the American Mathematical Society is already available; the user is invited to read the documentation of these packages, because they offer the user a large number of functionalities very useful for technical writings.

In typesetting mathematics the user should pay attention to the ISO rules concerning physics and technology (ISO-31/XI, 1978); s/he should pay a lot of attention to the various fonts that *must* be used for this kind of math. In short terms the ISO rules say the following.

1. Math italics must be used for all scalar variables and physical constants.
2. Serifed upright fonts must be used for function names, mathematical constants, and super- and sub-scripts that represent appositions to the variables.
3. Bold italics are used for matrices and vectors.
4. Bold roman fonts are used for sets, for which the double stroke blackboard bold, if available, may also be used.
5. Upright sans serif fonts are used to label objects in descriptive physical diagrams.
6. Oblique sans serif fonts are not used.
7. Upright bold sans serif fonts are not used.
8. Oblique bold sans serif fonts denote tensors.

Pay attention to Greek letters: by default pdfLaTeX uses oblique lower case letters and upright uppercase ones; ISO rules require them all to be oblique when they represent quantities; with LuaLaTeX and the option math-style=ISO to unicode-math this situation is corrected; with pdfLaTeX one may resort to the pm-isomath package that provides a lot of commands to use the proper family, series and shape for all symbols, not only the Greek ones. But observe the following three different uses of the symbols rendered with a Greek pi: $\mathrm{\pi}$ designates the transcendental math constant $3.141\,592\,653\,589\,793\ldots$; π represents the measurable quantity "flat angle"; $\mathsf{\pi}$ represents the physical particle "pion"; $\boldsymbol{\mathsf{\pi}}$ may represent a tensor. The pm-isomath package is incompatible with

[3] Sometimes the general solutions of the secondo degree equation refer to general coefficients, such as in $ax^2+bx+c = 0$. Elementary manipulation of the general equation, brings it to the form used here.

LuaLaTeX and is automatically skipped if the document is typeset with this program. Moreover the commands to address the various fonts properties and the single required glyphs are different from those used with unicode-math with the math-style=ISO option; see the documentation of both packages to find out the differences. See also the documentation of package isomath, (Milde, 2012); pm-isomath was created to override some isomath limitations; in facts the latter package provides full functionality only when using certain math fonts; pm-isomath supposedly works with any font collection, but with some compromises.

Of course there is more than the above short summary, which can be used as a simplified reminder that things are a little more complicated than what we might remember from our early studies.

For physicists and technologists (engineers) the ISO regulations forbid empirical equations between measures, because only the relations between quantities are admitted. Furthermore only the SI units are allowed and any other unit of measure is forbidden; therefore no CGSm, no CGSe, no CGS-Gauss; no British or American units. The use of the siunitx (Smith, 2018) package helps very much to use quantity equations, to typeset tables containing quantities, and so on.

One of the difficult things to remember is the fact that appositions to the quantity symbols must be typeset with roman fonts, while mathematical subscripts must be typeset with italic math fonts: therefore V_{max} is correct, while V_{max} is wrong; if V represents a voltage, V_{i} may denote an "input" voltage, while V_i represents the i-th element in an ordered set of voltages.

Package amsmath allows to typeset matrices, equation alignments; long expressions split at proper points in order to fit the measure, and many other features. For matrices see some examples in table 3.

Table 3: The six matrix types that can be created with the various commands made available by the amsmath package. The matrices in this table are built with the environments matrix, pmatrix, bmatrix, Bmatrix, vmatrix, Vmatrix

$$\begin{matrix}1&2&3\\4&5&6\\7&8&9\end{matrix} \qquad \begin{pmatrix}1&2&3\\4&5&6\\7&8&9\end{pmatrix} \qquad \begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$$

$$\begin{Bmatrix}1&2&3\\4&5&6\\7&8&9\end{Bmatrix} \qquad \begin{vmatrix}1&2&3\\4&5&6\\7&8&9\end{vmatrix} \qquad \begin{Vmatrix}1&2&3\\4&5&6\\7&8&9\end{Vmatrix}$$

File `toptesi-scudo-example.tex` contains some more examples. But I think that the best way to become an "equation expert" is to practice with all commands and environments described in the documentation of the amsmath package.

Beware: most office suits and some TeX system installations offer apps or dialog panes called "equation editors". They might appear useful at the very beginning, when the user is not yet comfortable with the myriad symbols and environments offered by the LaTeX kernel and the various extension packages. Such devices seduce the beginners, as well as they are seduced by editors like LyX that promise to display in the editor pane the result of a TeX system typesetting.

The above warning is to stress the point that LaTeX is a particular programming language that allows to define new functions and functionalities; those ready to use graphic user interfaces are prebuilt to do certain things while speeding up the graphic visualisation of the result. Equation editors and editors like LyX cannot do more than what they are programmed to do; actually modern versions of LyX have functions capable to translate TeX programming into LyX language; therefore if you want to use a specific LaTeX extension package, you can feed it to LyX so that it may translate the package code to its own language and save the result into a file that can be used in future instances. Time consuming, but LyX effective. But you cannot define your own commands to ease your own work, unless you prepare an extension package and submit it to the above mentioned LyX preprocessing. Why then using an editor that requires such preprocessing if any other editor works directly on your source file with no restrictions? Just to see in the editor pane something that resembles to what you hopefully would like to obtain from TeX? By experience I can say that in a long run it amounts in a waste of time, and it is not worth the amount of time you spend in preprocessing.

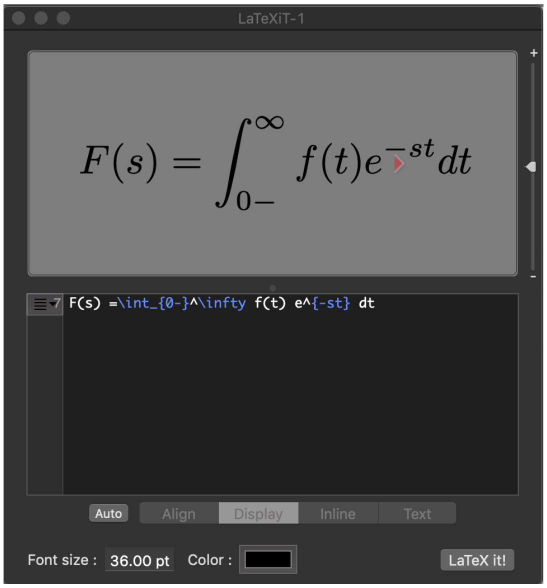
An example: figure 4 displays the editing pane of `LaTeXiT.app`, the equation editor installed on a Mac, when MacTeX, i.e. TeX Live for Mac, is installed.

As you see in the screenshot shown in figure 4 in the lower pane where you have to write LaTeX code; when you are finished, you click the bottom right button "LaTeXiT", and the app shows the result in the upper pane. Isn't that nice? Not quite: it recognises only the standard LaTeX commands, not the extensions of any package, and in particular the settings for fulfilling the ISO regulations that require a roman 'e' and a roman 'd' and suitable spacing before the integrator.

Even if the `LaTeXiT.app` is a fine piece of software, and the other equation editors generally behave more or less in the same way, don't use equation editors! You get stuck within their limits.

14.2 Figures

We already said that LaTeX can include graphic files in the formats PDF, EPS, and MPS (possibly vectorial), and JPG (lossy compressed bitmapped), and PNG (lossless compressed bitmapped). LaTeX



```
F(s) =\int_{0-}^\infty f(t) e^{-st} dt
```

Font size : 36.00 pt Color :

Figure 4: The graphic interface of the LaTeXit.app for Mac

can include drawings composed with its internal environments, and the various packages that allow advanced programmed drawing. There are also external programs that can produce output in the form of .tex files. At least one of these programs, asymptote, is installed together with TEX Live. This free software program is very useful for technical drawings; see for example (De Marco, 2009).

The user can install for example gnuplot, a mathematical drawing software: you enter a text file containing the necessary information about the functions to be drawn and the settings for the diagram, and the instructions for the output. With suitable output settings, the result is a .tex file that can be directly input into the user's thesis; see (Miklavec, 2013) to examine these functionalities; but you have to download the gnuplot program from its site https://sourceforge.net/projects/gnuplot/files/gnuplot/; there are versions for the three main platforms Windows, Mac, and Linux; but there are also the source files, so that it is possible to customise the executable program to the specific needs of the user's computer and operating system.

Another external free software program is Inkscape; it provides a graphical interface to draw almost anything; a nice part is that the labels that identify the various parts of the drawing are output as a separate file that superimposes the labels (typeset with the current fonts in the TEX output) on the vectorial PDF graphic file it produces. This property is highly desirable, and it is not very common that external programs can guarantee such functionality.

In facts some diagrams may be created from numerical tables set up with, say, Microsoft Excel and exported in PDF format. But as anybody can experience, the fonts used to label the axes and other similar text labels are clearly set with fonts that have nothing to do with those used to typeset the user's thesis.

The native LaTeX extension packages, such as the extended picture environment, the PSTricks bundle, the TikZ bundle, the pgfplots bundle, all create fine drawings while using the current fonts in the source file. I would suggest to keep in mind such extension packages, because the graphic output is certainly very professional and typographically correct.

Some years ago a frequent user of the Italian TUG forum, nicknamed Liverpool, opened a thread dealing with tracing of lossodromic and orthodromic routes on a 3D sphere projected on the plane together with meridians and parallels, and tracing those arcs in the background of the sphere with thinner lines than those in the foreground. I was very happy to participate in this program and eventually I wrote the .dtx file that documented the PGF/Ti*k*Z library used to trace some routes in order to see the different paths along the orthodromic vs. the lossodromic route[4] joining two given points. The library eventually contained also the specific commands to draw the two routes on a gnomonic projection and on a Mercator map (cylindrical projection). Liverpool eventually sent me back the .dtx file that I correctly signed only with his name; up to now this library is unpublished. I think it is instructive to see at least the results we reached.

Figure 5 displays a couple of examples; on the left the routes connecting New York and Moscow are displayed while on the right the corresponding routes joining New York with Bangkok. The code to draw one of the two maps is the following.

```
\documentclass{standalone}
\usepackage{pgfplots}
\pgfplotsset{compat=1.11}
\usetikzlibrary{quotes, rotte}

\begin{document}
\begin{tikzpicture}
\begin{axis} [x={(-0.866cm,-0.5cm)},
              y={(0.866cm,-0.5cm)},
              z={(0,1cm)},
              anchor=origin, at={(0,0)},
              disabledatascaling,
              hide axis]
% Draw the globe
\addplot3 [surf, z buffer=sort,
```

[4] Just to remember: the orthodromic route is the shortest path on a spherical surface that joins two given points; the lossodromic route is the path joining the given points such that the compass bearing remains constant; the orthodromic route, therefore, is an arc of a great circle, while the lossodromic one, unless the given points lay on a parallel, is an arc of a sort of spiral that winds on the spherical surface from one pole to the other.

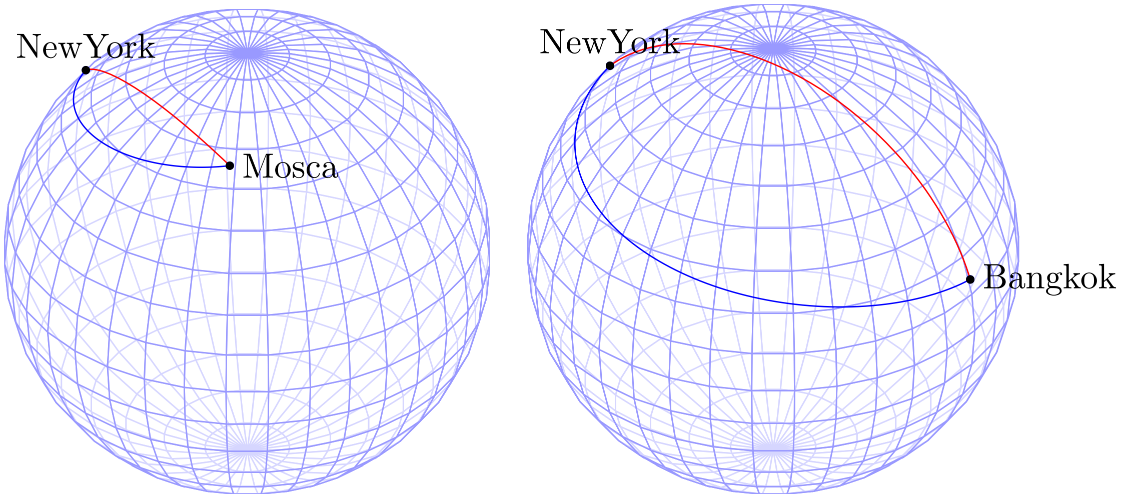


FIGURE 5: Two examples of routes joining two cities on the globe that are relatively close (New York and Moscow) or distant from one another (New York and Bangkok). The red route is the orthodromic one and the blue one is the lossodromic one.

```
           fill opacity=0.6, white,
           faceted color=blue!40,
           samples=19, samples y=37,
           variable=\u, variable y=\v,
           domain=0:180, y domain=0:360]
           ({2*cos(u)*sin(v)},
           {2*sin(u)*sin(v)}, {2*cos(v)});
% Draw orthodromic route
\pic [partenza={(40.744,-73.982)},
          arrivo={(13.725,100.51)},
          raggio=2cm, red] {ortodromia};
% Draw lossodromic route
\pic [partenza={(40.744,-73.982)},
         arrivo={(13.725,100.51)},
         raggio=2, smooth, samples=50,
         mark=*, mark size=1pt,
         blue] {lossodromia};
% Label departure point
\pic [posizione={(40.744,-73.982)},
        raggio=2, mark=*, mark size=1pt,
        "NewYork", above] {coordinata=A};
% Label arrival point
\pic [posizione={(13.725,100.51)},
        raggio=2, mark=*, mark size=1pt,
        "Bangkok", right] {coordinata=B};
\end{axis}
\end{tikzpicture}
\end{document}
```

Some picture qualifiers are in Italian, because Liverpool and I did not work out this Ti*k*Z library for international use; here I just changed the comment lines to English language. The interesting point about these drawings is that geographical coordinates of the departure and arrival points are just given in (fractional) degrees of latitude and longitude on the globe. This example shows quite well the power of the graphic packages that are part of any updated and complete TEX system installation. The code is not terribly complicated; the difficult part, if nothing else is already available, is to write the macros shown above and that may be saved in a Ti*k*Z library.

TABLE 4: The Smith family

	The Smith family		
Name	Role	Age	Activity
John	father	47	employee
Mary	mother	44	primary school teacher
Johanne	daughter	14	junior high student
Peter	son	8	primary school pupil

14.3 Tables

Tabular typesetting is possibly the most difficult task in typography. When typographers were using metal types, they would charge extra money for books and other printed material that contained tabular material. Of course with LATEX everything is much easier, but creating professional tables is still something that is out of our common experience; moreover, since most people create tables using word processors, and since this kind of software has limited performances, most of the time the tables we happen to read are typeset in a manner that is far from professional.

Of course a PhD thesis should have professional tables. Some tables are already shown in the mentioned `toptesi-scudo-example.pdf` where the same table is typeset in a non professional way, in a better way and in a professional way.

But here we should go into the details.

In table 4 you can see a small table typeset with the rules of the best typographical practice.

Notice the details.

1. The table does not contain any vertical rule.

2. The three horizontal rules have different meanings; the first and the last are thicker and delimit the table. The thinner middle rule separates the column headings from the other column cells. No other rules are necessary and even the first and last ones may be omitted. There are very rare occasions when a middle rule or a partial middle rule (spanning just a subset of columns) might add something to the table "meaning". LaTeX does not directly provide any means to fix the rule widths, but the (almost compulsory) booktabs package (Fear, 2016) comes to rescue; in practice these package facilities should be used for any table.
3. There is no need to emphasise the table headers with boldface or italic fonts, but it is not forbidden.
4. The cells contents are left or right justified within their cells; they might be also centred or form narrow paragraphs; in this last case it is better that the paragraph is typeset ragged right; LaTeX provides these facilities, but the array package helps very much with other paragraph-like cell contents and with very useful functionalities to customise whole columns or single cells.
5. Any table cell may contain another full table, i.e. the tabular environment may be nested.
6. In most countries, Italy included, the table caption is set over the tabular material as in table 4. In English speaking countries tables have their caption under the tabular material. There is no best practice, in the sense that both placements are correct in the proper country.
7. The example table natural width is too large compared to the column width, but not so large to suggest a top page centred full width float position. This is a common situation even if the document is being typeset in onecolumn mode. In this case the solution that I chose consists in reducing the font size; this is simple if the used fonts are continuously scalable, or better, if they are piecewise continuously scalable because they have available optical sizes. In this paper, typeset with the Latin Modern font collection that has optical sizes, the chosen solution appears to be the most comfortable one. Characters of size 8.5 pt are too small to be read? Yes, may be; in this case another solution might be to reduce the inter-column whitespace width; by default it is 12 pt wide, approximately 4 mm; there is enough space to reduce the whole table without reducing the font size too much. See then table 5 where the font size is 9.5 pt and the inter-column width is just 6 pt.

Table 5: The Smith family

The Smith family			
Name	Role	Age	Activity
John	father	47	employee
Mary	mother	44	primary school teacher
Johanne	daughter	14	junior high student
Peter	son	8	primary school pupil

The above small list describes the best practice, but it also underlines the big or small problems that come up with tables. By typesetting in onecolumn mode, as the PhD ScuDo style requires, some of the described problems may vanish or are reduced. One suggestion I can give for tables that are too wide, is to use the widetable (Beccari, 2018) package with its widetable environment that computes the necessary inter-column width in order to fit into a specified width; of course, should it compute a negative inter-column width, it issues a warning and typesets the tabular material with the default value.

In other circumstances the solutions might be of different nature; one is to use the X column descriptors that produce paragraph-like cell contents, but their width is automatically computed to fit the specified table width.

Another solution is to typeset a definitely too large table in a sideways mode; the best practice requires to rotate the sideways material 90° counterclockwise independently of the fact that this material falls on an even or odd page. Several packages available for such rotations perform it in such a way that the caption base is also facing the outer trim margin. This is why I am not naming any particular package.

All "regular" tables are not broken across page breaks; since they are large objects, they need to be floated, so that LaTeX can find the best place to output them. LaTeX must fulfil certain conditions to output all kind of floats: how tall are they? how many at maximum may be in the same page? in which position on the page: top, bottom or within the text? how much residual space remains on the page for regular text? when in twocolumn mode should the float be in some position of a column or should it span the whole text block width? Actually this may be decided by the user, but again, depending on the type, the specific constraints similar to those for onecolumn mode remain valid. Such constraints are somewhat released when the float stack still contains some floats, but the chapter is finished and they must be all output before a new chapter starts; or when the end of the document is reached.

But if a table is definitely too long it may be typeset on several consecutive pages; the environments longtable and supertabular can do the job; I feel more comfortable with longtable, but it is just a question of personal taste: their performances are almost the same. Provided they are not wider than the text block height, there are also packages to typeset long tables in a sideways manner, continuing to use the same rotation, page after page, as far as the end of the table. I do not like such sideways long tables, but I understand that in certain circumstances it is impossible to avoid them. I think it is better to plan their construction in a smarter way. Since automatic procedures cannot be smarter than a human, humans should not rely on the limited intelligence of those procedures.

15 Bibliography

The bibliography is an important part of a doctoral thesis; it is not meant to document every piece of information that appears in the thesis. It should list the documents effectively read, the references that mostly describe the state of the art, not only in the discipline where the thesis may be classified, but also in side disciplines that were used during the development of the thesis.

A bibliography should not emphasise the visited Internet sites, even if some information or some data were taken from those sites; the explanation is simple; even if they do not transmit superficial or even fake data, they are not reliable; today they are accessible; tomorrow they might have disappeared from the Web. If the thesis is read by someone else in, say, ten years from now, it is very likely that more than a half of those sites will not exist anymore.

This is a common experience, therefore I avoid citing material that is not recorded in a stable way; in this paper I cite TeX related documentation because I assume that if you, the reader, are using the TeX system, you have the references that I cite on board of your computer.

15.1 The bibliographic database

The first step to create a bibliography is to create a database where each record describes everything connected with each reference in a formal way. There are at least two programs that ease this task.

1. The external program `JabRef` is a program that runs in a Java virtual machine; it is suitable for any operating system provided it has the Java bundle installed; `JabRef` is fully compatible with the TeX system. The record names are sort of standard and the descriptors of the various fields of each reference are consistent and standard; if one uses non standard descriptors, they are simply ignored together with the description they address. The program has its own graphical interface.
2. The external program `BibDesk` is only for Mac platforms and is already available when TeX Live is installed with MacTeX. This program has its own graphical interface.
3. Actually there is a hard way to create a bibliographic database: it consists in using the same text editor that is being used to handle the `.tex` files. The user can use it to create a `.bib` file containing the textual material of each reference record; of course s/he should have a clear understanding of the mandatory and the auxiliary information that any type of reference requires. The first two methods are much superior because those programs can fill in the mandatory descriptors for every document type and let you add any other descriptor or field that you find useful, even if it will not migrate to the final typeset bibliography.
4. It is strongly recommended to name the database with the same name as the main thesis file, but, of course, with the mandatory extension `.bib`. It is possible to enter a comma separated list of database files into the argument of `\addbibresource`. A thesis should not contain hundreds of references, so that it is not necessary to split a huge single database in a number of smaller chunks; sometimes it may be useful to keep the list of URLs, the books, the articles, the internal reports, and so on in separate databases. In this case extreme care should be paid to the fact that the citation keys must be unique among the whole set of databases.

An example of bibliographic database is the following.

```
@manual{man:ReferenceManual,
        Author = {Claudio Beccari},
        Address =
        {\url{http://www.guitex.org/home/images
            /doc/GuideGuIT/}},
        Organization = {{\GuIT}},
        Title = {{Il \LaTeX\ Reference Manual
                  commentato}},
        Year = {2017}}

@misc{misc:toptesi,
        Author = {Claudio Beccari},
        Howpublished = {{PDF document}},
        Note = {in
         {\url{$TEXMF/doc/latex/toptesi/
                toptesi.pdf}},
         \url{$TEXMF/doc/latex/toptesi/
               toptesi-it.pdf}},
        Title = {{La classe \pack{TOPtesi}}},
        Year = {2019}}

@book{book:Bringhurst,
        Address = {{Vancouver, BC}},
        Author = {Robert Bringhurst},
        Publisher = {Hartley \& Marks},
        Title = {The elements of typographic
                 style},
        Year = {2004}}

@article{art:Caignaert,
        Author = {Christophe Caignaert},
        Journal = {\textsl{TUGboat}},
        Number = {3},
        Pages = {161-174},
        Title = {A story of \textit{kpfonts}},
        Volume = {31},
        Year = {2010}}

@manual{man:ShortMathGuide,
        Address = {Providence, Rhode Island},
        Author = {Michael Downes},
        Edition = {1.09},
        Note =
        {In \url{ftp://ftp.ams.org/pub/tex/
           doc/amsmath/short-math-guide.pdf}},
        Organization = {American Mathematical
                        Society},
        Title = {Short math guide},
        Year = {2002}}


@manual{man:Flynn,
        Author = {Peter Flynn},
        Note = {In {\url{CTAN/tex-archive/info/
               beginlatex/beginlatex-3.6.pdf}}},
        Title = {Formatting information ---
                 A beginner's introduction
                 to typesetting with {\LaTeX}}}

@manual{man:xetex-companion,
        Author = {Michel Goossens},
        Note = {in \url{http://xml.web.cern.ch/
                XML/lgc2/xetexmain.pdf}},
        Organization = {{\LaTeX\, Team}},
        Title = {The {\XeTeX} Companion --
        {\TeX\ meets OpenType and Unicode}},
        Year = {2011}}
```

As it can be seen, every record starts with an "at sign" (@), followed by the name of a type of reference; most of these names are full words, but some are simple abbreviations, such as misc (miscellaneous) to be used when the reference is difficult to be classified.

Each record content is enclosed within balanced braces; it is divided in fields and each field has a name followed by the "equals sign", followed by a balanced braces pair containing the field contents; the only exception is the first field that contains the citation key without any name; each field content is actually separated from the next one with a comma. Except for the Author field, all other fields, if they were not enclosed within balanced braces, are transformed to lowercase except for the first word initial letter. This is why it is better to always enclose the field contents within braces. This lowercasing would take place also with LaTeX macros; therefore during typesetting those lowercased macros become typesetting errors of the type "Undefined control sequence"; tricky error, because when the source .bib file is examined no errors are found. But if the processed bibliography file with extension .bbl is examined the error is immediately spotted.

There are at least two ways to typeset a bibliography from a bibliographic database that depend on the bibliography external processor bibtex or biber, and on the type of bibliography style chosen.

The ScuDo doctoral thesis is typeset trough the use of the biber processor, therefore here nothing is said concerning the use of bibtex.

The toptesi-scudo.sty module presets the bibliography processing through the biblatex package (Lehman, 2018) and its options in order to get an alphabetically ordered bibliography on the basis of the author's surname, but with each reference identified by a number. The setting is as this:

```
1 \unless\ifmybibstyle
2   \usepackage[autostyle]{csquotes}
3   \usepackage[backend=biber,
4               style=numeric-comp,
5               citestyle=numeric,
6               sorting=nty,
7               natbib]{biblatex}
8   \addbibresource{\jobname.bib}
9 \fi
```

The test starting on line 1 and completed on line 9 preloads the necessary packages and sets the options to describe the bibliography style only if the class option mybibliography was *not* specified by the user. If it was, nothing is done and the user becomes totally responsible to select the procedure and the style s/he prefers *if and only if s/he got an authorisation from his/her supervisor*. This is because the style of the ScuDo doctoral theses should be conformant to a model that lets anyone recognise at first sight the school theses.

On line 2 the csquotes package (Lehman and Wright, 2018) is loaded; this allows to adapt the quotation marks to the style used in the country where a specific language is used. Actually the records of the bibliography database might contain a field that qualifies the reference as one to be written in a certain language; this is useful also for the correct word hyphenation at line breaks.

Then from lines 3 to 7 the biblatex is loaded with the necessary options; specifically:

backend=biber specifies that the biber processor is used to extract and format the various references.

style=numeric-comp specifies that the numeric identification of every reference is used.

citestyle=numeric the same style is specified for citations.

sorting=nty specifies that the sorting is principally based on author names; for equal names, sorting is based on titles; for equal titles, sorting is based on year of publication.

natbib allows to use the same citation schemes established by the natbib package (Daly, 2010).

Line 8 specifies that the bibliographic database file is named \jobname.bib. In his/her thesis the candidate may specify the names of other bibliographic databases with similar commands; the bibliography processor will examine the whole set in order to extract and format the various entries. The important point is that entries in all databases have a unique citation key.

With the default settings the bibliography turns out as in figure 6, while with the settings used in the sample file `toptesi-scudo=example.tex` the bibliography turns out as in figure 7.

All this implies that the `biber` program must be run after at least one compilation of the thesis and after any modification of the databases; but it is not necessary to run it again and again after each compilation of the thesis.

16 Nomenclature

Personally I find a glossary or a nomenclature list, a list of symbols, a list of acronyms, whatever you want to call them, useful just in certain circumstances. Of course a purist may distinguish those lists for many details. But they have something in common. All of them are lists of lexemes, and of each one they explain what they mean or represent.

It is possible that in a doctoral thesis some words appear to be used in a special or restricted way and are not commonly known even to scholars in the disciplinary field. I assume that such terms are very rare and that the examining committee is made up of experts. For the members of this committee a nomenclature list should be superfluous.

Nevertheless it is possible that in the future the thesis is read by people that are not so expert. For these people a nomenclature list might be very useful.

The tipotesi=scudo already loads the nomencl package; and already defines some categories of names to be described. The example file `toptesi-scudo-example.tex` makes some examples of nomenclature entries (with silly definitions, just to show how to use them); but the real bonus is that this TOPtesi module already contains the shell-escape commands necessary to typeset the nomenclature list just in one run. In facts in a "normal" situation the user should enter with similar commands into the main or one of the secondary files the material that forms the nomenclature list; s/he should compile the thesis, then s/he should open a terminal or command prompt, and should enter the necessary operating system command to run the external program `makeindex` that processes the TEX material that the typesetting program already transferred to a raw nomenclature file; the result of this external processing is a definitive `.nls` file, that in the next typesetting run will become the typeset nomenclature. It is more complicated to describe it than to do it; but even so, everything is already hardcoded into the ScuDo module, so that the user should not care about anything related to the nomenclature typesetting.

17 Index

The implemented process to create the nomenclature is similar to the one used for creating an index, but in this case the whole machinery is already included into the imakeidx package that is preloaded by the ScuDo module.

With indices the problem is more complicated than for nomenclatures, because the user might desire to make more than one index at the same time.

Whatever is listed in an index is generally followed not by a description as in a nomenclature, but by the list of the pages in the thesis where something interesting is said concerning each lemma. The page numbers might be typeset with different fonts as it is done in the TEX book; where a boldface page number is used to point to the page where there is the lemma definition, a normal upright page number is where the lemma is used in a significant way; an italic page number may represent the page where the lemma is used in an application, and so on.

During the typesetting task the material to appear in the index or indices is collected and written in one or more raw `.idx` files; before the typesetting task is terminated the written files are transmitted through operating system commands to the external processing program that produces one or more ordered index files; then, always before terminating the typesetting task the processed files are input into the main typesetting flow and the index or indices are typeset. The whole procedure is very handy and I do not use anything else to produce the indices I need. And I think that this way of processing indices and nomenclatures is the most useful one.

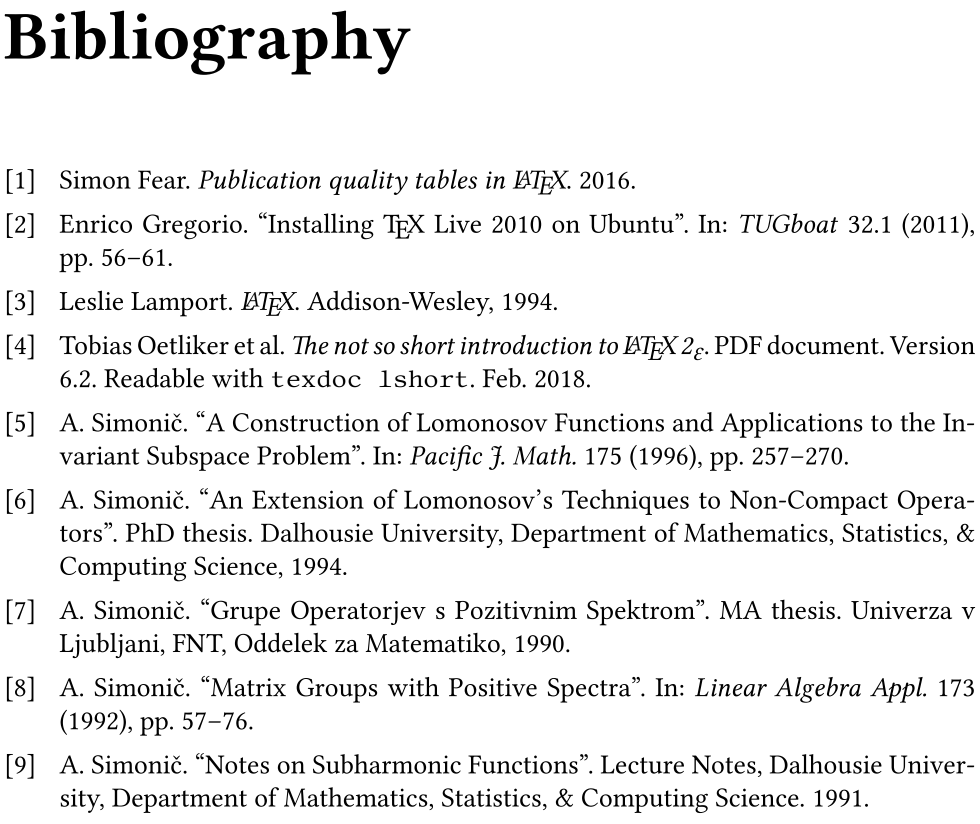
It is assumed, of course, that nomenclature and indices appear at the very end of the typeset thesis, therefore after the appendices and the bibliography.

17.1 Configuring `\makeindex`

The normal use of the indexing facilities require that the preamble of the document contains the `\makeindex` declaration. With package imakeidx this declaration may be customised:

```
\makeindex[%
  ⟨index name⟩,
  ⟨index title⟩,
  ⟨other options⟩]
```

The ⟨*index name*⟩ is a symbolic name to distinguish which index should be activated with the specified options; if this option is omitted, the settings apply to the default index. This optional ⟨*index name*⟩ is also used to configure the `\index` command to send its argument to the specified index.



Bibliography

[1] Simon Fear. *Publication quality tables in LaTeX*. 2016.

[2] Enrico Gregorio. “Installing TeX Live 2010 on Ubuntu”. In: *TUGboat* 32.1 (2011), pp. 56–61.

[3] Leslie Lamport. *LaTeX*. Addison-Wesley, 1994.

[4] Tobias Oetliker et al. *The not so short introduction to LaTeX 2_ε*. PDF document. Version 6.2. Readable with `texdoc lshort`. Feb. 2018.

[5] A. Simonič. “A Construction of Lomonosov Functions and Applications to the Invariant Subspace Problem”. In: *Pacific J. Math.* 175 (1996), pp. 257–270.

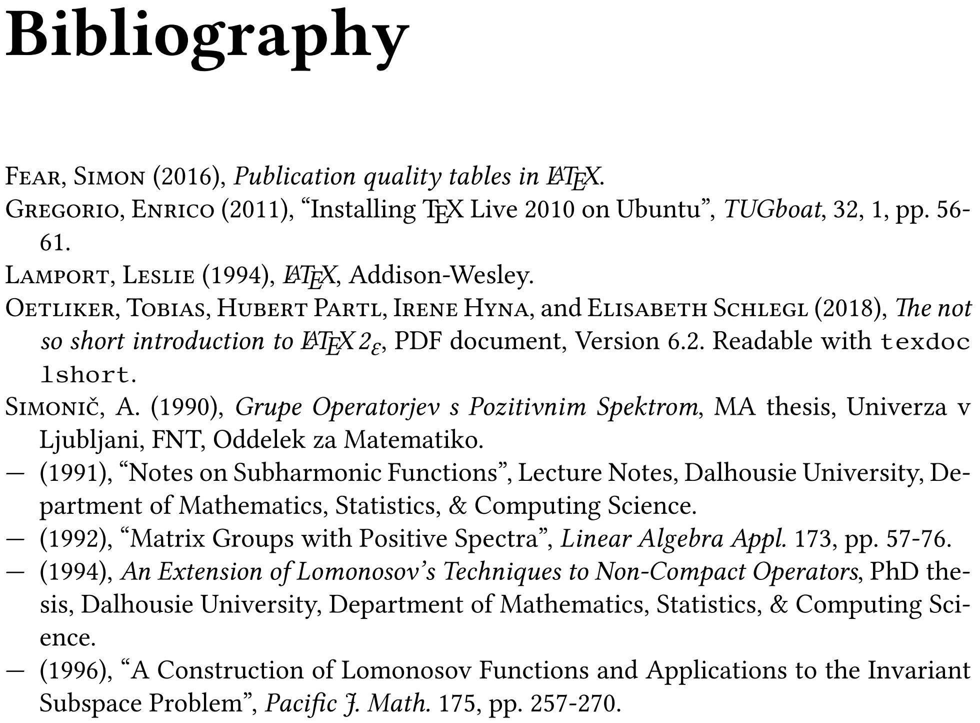
[6] A. Simonič. “An Extension of Lomonosov’s Techniques to Non-Compact Operators”. PhD thesis. Dalhousie University, Department of Mathematics, Statistics, & Computing Science, 1994.

[7] A. Simonič. “Grupe Operatorjev s Pozitivnim Spektrom”. MA thesis. Univerza v Ljubljani, FNT, Oddelek za Matematiko, 1990.

[8] A. Simonič. “Matrix Groups with Positive Spectra”. In: *Linear Algebra Appl.* 173 (1992), pp. 57–76.

[9] A. Simonič. “Notes on Subharmonic Functions”. Lecture Notes, Dalhousie University, Department of Mathematics, Statistics, & Computing Science. 1991.

Figure 6: Bibliography with numerical labels



Bibliography

Fear, Simon (2016), *Publication quality tables in LaTeX*.

Gregorio, Enrico (2011), “Installing TeX Live 2010 on Ubuntu”, *TUGboat*, 32, 1, pp. 56-61.

Lamport, Leslie (1994), *LaTeX*, Addison-Wesley.

Oetliker, Tobias, Hubert Partl, Irene Hyna, and Elisabeth Schlegl (2018), *The not so short introduction to LaTeX 2_ε*, PDF document, Version 6.2. Readable with `texdoc lshort`.

Simonič, A. (1990), *Grupe Operatorjev s Pozitivnim Spektrom*, MA thesis, Univerza v Ljubljani, FNT, Oddelek za Matematiko.

— (1991), “Notes on Subharmonic Functions”, Lecture Notes, Dalhousie University, Department of Mathematics, Statistics, & Computing Science.

— (1992), “Matrix Groups with Positive Spectra”, *Linear Algebra Appl.* 173, pp. 57-76.

— (1994), *An Extension of Lomonosov’s Techniques to Non-Compact Operators*, PhD thesis, Dalhousie University, Department of Mathematics, Statistics, & Computing Science.

— (1996), “A Construction of Lomonosov Functions and Applications to the Invariant Subspace Problem”, *Pacific J. Math.* 175, pp. 257-270.

Figure 7: Author-year bibliography

Again the ⟨*index title*⟩ is useful to set a title to each of several index files and to use it in the printed indices. Of course, even if just one index is created, it is possible to give it a title different from the default one.

The ⟨*other options*⟩ refer to other details of each index: for example it is possible to typeset an index in a number of columns different from the twocolumn default mode. The user is invited to read the documentation of the imakeidx package documentation (Gregorio, 2016).

17.2 Entering data to an index

Entering each entry to every index is done through the `\index` macro. This is a very particular macro: first, it is disabled when `\makeindex` declaration has not been specified; second, the macro argument has a very special syntax, because it is fed to the external program `makeindex` that requires that syntax. The original documentation by Leslie Lamport Lamport (1987) dates back to 1987, but with minor variants is still valid. A general document on making indices (Chen and Harrison, 2014) is more recent because it is dated 2014, but still it does not mention the enhancements provided by package imakeidx.

The general syntax for entering index data is the following (to be input in just one line):

⟨*first level*⟩`!`⟨*second level*⟩`!`⟨*sort entry*⟩
`@`⟨*typeset entry*⟩`|`⟨*address*⟩

Where ⟨*sort entry*⟩ is the string used to sort the index, while ⟨*typeset entry*⟩ is what appears in the index; for example, the user wants to enter the word "Transistor" in boldface, but it does not want to sort it among the capitalised entries: then ⟨*sort entry*⟩ will be "transistor" and the ⟨*typeset entry*⟩ will be "`\textbf{Transistor}`". Moreover ⟨*address*⟩ is the way to typeset the page number, or the string to use so as to point to another entry. The ⟨*first level*⟩ entry and ⟨*second level*⟩ entry are used only when the ⟨*typeset entry*⟩ should appear hierarchically under other entry levels.

It is convenient to define handy macros that use the correct syntax; for example in this paper I use the command `\pack` to enter package names in upright sans serif font: if I wanted to create an index containing the package names grouped under a first level entry "packages", in the preamble I would define the following code (that requires the use of the xparse package, The LaTeX 3 Project Team (2018)):

```
\newcommand\packagestyle[1]{\textsf{#1}}
\NewDocumentCommand\pack{s m}{%
\packagestyle{#2}\IfBooleanTF{#1}{}%
 {\index{packages!#1@\packagestyle{#2}}}%
}
```

The first definition establishes how I want to typeset package names; the second definition (that requires the xparse facilities) defines a command `\pack` that accepts an optional star and a mandatory name of a package. If the first optional argument is a star, the command just typesets the package name, otherwise, besides typesetting the package name in the document, it sends the indexing information to the output `idx` file; the external program `makeindex` (automatically invoked by imakeidx) creates an entry "packages" and a subentry with the name of the package typeset with the proper font. Therefore I have available two ways of using the command `\pack`:

- `\pack*{`⟨*package name*⟩`}` to simply typeset the ⟨*package name*⟩ in the document; and
- `\pack{`⟨*package name*⟩`}` to typeset the ⟨*package name*⟩ in the document and simultaneously send the subentry ⟨*package name*⟩ to the index.

If the user wishes to create more than one index, in the preamble of the document s/he should customise more than one `\makeindex` declaration, for example

```
\makeindex[intoc]
\makeindex[name=places,
           title=List of places,
           intoc, columns=1]
```

By so doing the user sets two indices, the first one is the normal one with the default name and it is listed in the table of contents. The second, named `places`, has the title "List of places"; it is typeset in onecolumn mode; it is listed in the table of contents.

In the body of the document, the user introduces such commands as in this text example:

```
Albert Einstein was born in
Ulm\index[places]{Germany!Ulm@Ulm} and
while he was working in
Bern\index[places]{Switzerland!Bern@Bern}
started working on the theory of
restricted relativity%
\index{relativity!restricted@restricted}.
```

The mechanism is pretty simple to be used; and it becomes simpler if the user defines suitable macros to enter the necessary information just once; the xparse package has facilities to define macros that accept almost any kind of mandatory and optional arguments. The difficult part of making indices remains the author choice of which lemma instances s/he wants to mark with the `\index` command.

18 Archivable format

Politecnico di Torino requests a copy of each thesis to be submitted to the registrar when each student submits the formal application in order to defend his/her thesis. These copies must be "dematerialised", i.e. in PDF format, provided that this PDF file fulfils the regulations issued by the International Standards Organisation (ISO) for long term archivability. Such ISO regulations were first issued in 2005, and in the following years they received further additions. The documentation of the pdfx package, (C.V. Radhakrishnan *et al.*, 2018), describes the various levels of ISO conformity and describes how to obtain conformant PDF files by using the package.

The LaTeX based typesetting engines that are part of the TeX system up to now cannot produce files with tagged contents, named *Tagged PDF* files. This means that today such engines can produce PDF files that are conformant only to the PDF/A-1b regulations. In another paper Ulrike Fisher is going to describe the progress in modifying these engines in order to produce tagged PDF files, and therefore files that can be conformant with the more stringent regulation PDF/A-1a and also to other more recent standards.

The constraints a PDF/A-1b conformant document must fulfil deal with some delicate points.

PDF level Even the PDF language underwent to several upgrades so that the ISO established for PDF/A-1b files that the PDF language level should be exactly 1.4, no less, no more. More recent ISO standards allow higher language levels.

Fonts The fonts used in the document must be vectorial; bitmapped fonts are absolutely forbidden. Glyphs with zero width are forbidden. Unfortunately some standard TeX math fonts do not fulfil such constraint. The TOPtesi bundle contains a patch to this problem, but actually there is no need to do any patch when the typesetting engine is LuaLaTeX and Unicode math fonts are used.

Encoding UNICODE and certain TrueType fonts are accepted; Type 1 fonts are accepted only if the file with the correspondence of their glyph addresses with UNICODE is included. For this reason it is better to avoid Type 1 fonts and therefore it is necessary to typeset the PDF document with LuaLaTeX. It would be possible to use also XeLaTeX, but special postprocessing would be required.

Colors Colors should be only RGB (and/or grayscale, but let us forget this color code, since it is already covered by the RGB profile).

Color profile The used color profile must be included within the mandatory metadata.

Metadata Special metadata must be included and they must not be compressed within the PDF file, so that they are always readable without uncompressing the file.

Dublin Agreement data The metadata concerning the Dublin Agreement have to be included in the proper form.

It is evident that the requirements are pretty stringent even for the less stringent regulation among the various levels of the ISO rules.

With the ScuDo doctoral school dissertation, things are pretty safe, because the preamble example contained in the `toptesi-scudo-example.tex` guarantees a very high success in producing PDF/A-1b compliant files.

The preamble of the document should be as follows[5]:

```
\documentclass[%
   corpo=12pt, % font size
   twoside, % recommended
   tipotesi=scudo,
]{toptesi}
...
\begin{filecontents*}{\jobname.xmpdata}
\Author{Mario Rossi}
\Title{Writing Your Ph.D. Thesis
       with LaTeX}
\Subject{Doctoral dissertations
       in the SCUDO doctoral school}
\Keywords{PDF\sep
          PDF/A\sep
          ISO 19005\sep
          LaTeX\sep
          PhD Thesis\sep
          Engineering\sep
          SCUDO}
\Publisher{Politecnico di Torino}
\end{filecontents*}
...
\usepackage[a-1b]{pdfx}
\ifPDFTeX
  ...
\else
  \usepackage{fontspec}
  \defaultfontfeatures{Ligatures=TeX}
  \setmainfont{Libertinus Serif}
  \setsansfont{Libertinus Sans}
  \setmonofont{Libertinus Mono}%
              [Scale=MatchLowercase]
  \usepackage[math-style=ISO]{unicode-math}
  \setmathfont{Libertinus Math}%
\fi
...
\makeindex[intoc]% configure indexing

\unless\ifcsname ver@hyperref.sty\endcsname
  \usepackage{hyperref}\fi
\hypersetup{%
  pdfpagemode={UseOutlines},
  bookmarksopen,
  pdfstartview={FitH},
  colorlinks,
```

[5] The example file contains a lot of descriptive comments; for brevity here the comments are neglected as well as unnecessary options and commands.

```
46   linkcolor={blue},
47   citecolor={blue},
48   urlcolor={blue}
49 }
50 \includeonly{%
51 ...
52 }
53 %
54 \begin{document}
```

The toptesi-scudo module is conceived in such a way as to preload all the necessary packages; in particular the filecontents package (Pakin, 2018) is preloaded for convenience. In facts the starred or unstarred environment filecontents that is used in lines from 7 to 21 in the above code, normally would not overwrite the file it is supposed to create; therefore, in order to change the environment contents even for the simple correction of a typo, it is necessary to delete the already existing `.xmpdata` file, otherwise the correction does not appear in the output file. With the use of the filecontents package this feature is eliminated and the file is overwritten each time the document is typeset.

The environment output file `\jobname.xmpdata` has this special name that assures that the generated file has the same name as the thesis main file and the specified correct extension.

This file contains the metadata that depend on the specific document; all other necessary metadata are fixed information provided by the pdfx package invoked with the a-1b option which specifies the document PDF file should be conformant with the PDF/A-1b regulation.

The `.xmpdata` file contains certain metadata, each one preceded by its keyword; the metadata concerning that example document contain the name of the `Author` as “Mario Rossi” (the Italian equivalent of the Anglo-American John Smith); the main title for the `Title` descriptor and a short but descriptive phrase for the `Subject`; some `Keywords` are also listed separated by the special separator `\sep` required by package pdfx. Of course more specific metadata with other descriptors may be specified; the whole list is found in the documentation of package pdfx.

The lines from 24 to 35 contain the test to discover if the typesetting engine is `pdftex` or something else. We already discussed this feature in subsection 12.1 and here we skip what deals with pdfLaTeX.

As the reader can observe, package pdfx is loaded as the very first one (besides those that are already preloaded); this package loads hyperref (Rahtz and Oberdiek, 2018) with the option pdfa so that certain features of hyperref are modified in order to fulfil the hyperlinking requirements set forth by the iso rules; this is why in lines 39 and 40 the preamble tests if package hyperref has already been loaded and in case it loads it *without any option*. Such options are set with the command `\hypersetup` that assures to avoid any conflict between the options passed to the package by pdfx and this second possible call. Option clashes produce error messages difficult to correct; so it is better to avoid them in advance.

The text and math fonts loaded with the fontspec facilities are those called Libertinus, a revisited set of OpenType fonts obtained from the Libertine ones; apparently the latter are not maintained any more, and Michael Sharpe reworked, corrected and enriched them and called them with a different, although similar, name; they lack the optical sizes but do a very good job. The math version must be specified after the unicode-math package has been loaded; here I specified the math-style=ISO option because I recommend it very strongly for the ScuDo doctoral theses in engineering disciplines.

19 Conclusion

In this paper I described the problems that arise when typesetting theses in general and how most of these problems have been tackled with the TOPtesi bundle. I entered into the details of the module needed to create typographically pretty nice theses for the ScuDo Doctoral School of Politecnico di Torino. In any case the requirements set forth by the School are all satisfied; all necessary macros and environments have been defined and are available to the authors.

The authors may introduce more macros and environments in order to create their theses. But what is really important, I wish them to be proud of their researches and their results; they will be proud also of the appearance of their typeset results.

References

Ahmetovic, D., T. Armano, M. Berra, C. Bernareggi, A. Capietto, S. Coriasco, N. Murru and A. Ruighi (2018). «Axessibility: creating PDF documents with accessible formulae». *ArsTEXnica*, (26), pp. 50–54.

American Mathematical Society (2013). *The amssymb package*. Version 3.01. Readable with `texdoc amssymb`.

— (2017). *Using the amsthm package*. Version 2.20.3 – Readable with `texdoc amsthm`.

— (2018). *User’s guide for the amsmath package*. Readable with `texdoc amsmath`.

Beccari, Claudio (2018). *The widetable package*. GuIT. Version 1.5. Readable with `texdoc widetable`.

— (2019a). *Il pacchetto TOPtesi*. GuIT. Version 6.2.03. Readable with `texdoc toptesi-it`.

— (2019b). *The extension package curve2e*. GuIt. Version 1.61. Readable with `texdoc curve2e`.

— (2019c). *The TOPtesi bundle*. GuIt. Version 6.3.02. Readable with `texdoc toptesi.pdf` and in Italian `texdoc toptesi-it`.

Carlisle, David (1995). *The indentfirst package*. TUG. Readable with `texdoc indentfirst`.

Chen, Pheong and Michael A. Harrison (2014). «Index preparation and processing». PDF document. The original document is older; this is an updated version by Dan Lueking and Karl Berry. Readable with `texdoc ind`.

C.V. Radhakrishnan, Hàn Thế Thành, Ross Moore and Peter Selinger (2018). *Generation of PDF/X- and PDF/A- compliant PDFs with* `PdfTeX`. River Valley Technologies, Trivandrum, India. Version 1.6 upgraded and valid also for LuaLaTeX. Readable with `texdoc pdfx`.

Daly, Patrick W. (2010). *Natural sciences citations and references — Author-year and numerical Schemes*. TUG. Version 8.31b. Readable with `texdoc natbib`.

De Marco, Agostino (2009). «Produrre grafica vettoriale di alta qualità programmando asymptote». *ArsTeXnica*, (8), pp. 25–39.

Fear, Simon (2016). *Pubblication quality tables in LaTeX*. TUG. Version 1.618033. Readable with `texdoc booktabs`.

Feuersänger, Christian (2018). *Manual for package* pgfplots. TUG. Version 1.16. Readable with `texdoc pgfplots`.

Gässlein, Hubert, Rolf Niepraschk and Josef Tkadlec (2016). *The pict2e package*. TUG. Version 0.3b. Readable with `texdoc pict2e`.

Gregorio, Enrico (2016). «The package imakeidx». PDF document. Version 1.3e, Readable with `texdoc imakeidx`.

Hammmerlindl, Andy, John Bowman and Tom Prince (2018). *Asymptote: the vector graphics language*. TUG. Version 244. Readable with `texdoc asymptote`.

Hobby, John D. (2018). *METAPOST– A users' manual*. TUG. Version 2.00 (2.0rc2). Readable with `texdoc metapost`.

ISO 19005-1 (2005). *ISO 19005-1:2005 – Document management – Electronic document file format for long-term preservation*. International Organization for Standardization, Geneva. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38920`.

ISO 19005-2 (2011). *ISO 19005-2:2011 – Document management – Electronic document file format for long-term preservation*. International Organization for Standardization, Geneva. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50655`.

ISO 19005-3 (2012). *ISO 19005-3:2012 – Document management – Electronic document file format for long-term preservation*. International Organization for Standardization, Geneva. `http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=37&ics2=100&ics3=99&csnumber=57229`.

ISO 31/XI (1978). *Mathematical signs and symbols for use in the physical sciences and technology*. International Organization for Standardization, Ginevra. Regulation ISO 31/XI/–1978. Updated with regulation ISO 80000-2: `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31887`. See also (Thompson and Taylor, 2008).

Lamport, Leslie (1987). *MakeIndex: an Index Processor for LaTeX*. TUG. Readable with `texdoc makeindex`.

— (1994). *A document preparation system — LaTeX — User's guide and reference manual*. Addison Wesley, Reading, Mass., 2^{nd} edition.

Lehman, Philip (2018). *The biblatex package – Programmable bibliographies and citations*. TUG. Version 3.12. Readable with `texdoc biblatex`.

Lehman, Philip and Joseph Wright (2018). *The csquotes package — Context sensitive quotation facilities*. TUG. Version 5.2d. Readable with `texdoc csquotes`.

Miklavec, Mojca (2013). *Using* `context` *and tikz terminals for gnuplot in ConTeXt*. Readable with `texdoc gnuplot`. This document gives insructions to install the external program `gnuplot`, version 4.6.0 or later, and to configure it to be used by the typesetting programs of the TeX system.

Milde, Günter (2012). *isomath – Mathematical style for science and technology*. TUG. Readable with `texdoc isomath`.

Oetliker, Tobias (2015). *An acronym environment for LaTeX 2_ϵ*. tug. Verison 1.41. Readable with `texdoc acronym`.

Pakin, Scott (2018). *The filecontents package*. TUG. Version 1.4. Readable with `texdoc filecontents`.

Rahtz, Sebastian and Heiko Oberdiek (2018). *Hypertext marks in LaTeX: a manual for hyperref.* TUG. Readable with `texdoc hyperref`.

Robertson, Will (2019a). *Experimental Unicode mathematical typesetting. Th unicode-math package.* TUG. Version 0.8n. Readable with `texdoc unicode-math`.

— (2019b). *The fontspec package. Font selection for XeLaTeX and LuaLaTeX.* TUG. Version 2.7b. Readable with `texdoc fontspec`.

Smith, Joseph (2018). *siunitx — A comprehensive (SI) units package.* TUG. Version 2.7s. Readable with `texdoc siunitx`.

Talbot, Nicola L.C. (2019). *User manual for glossaries.sty v4.42.* TUG. Version 4.42. Readable with `texdoc glossaries`.

Tantau, Till (2019). *TikZ & PGF.* TUG. Version 3.1.1 – Readable with `texdoc tikz` or with `texdoc pgf` or with `texdoc pgfmanual`.

The LaTeX 3 Project Team (2018). *The xparse package – Document command parser.* TUG. Readable with `texdoc xparse`.

Thompson, Ambler and Barry N. Taylor (2008). *Guide for the Use of the International System of Units (SI).* NIST – National Institute of Standards and Technology. NIST Special Publication 811 – 2008 Edition. Url: `https://www.nist.gov/pml/special-publication-811-extended-contents`.

Veytsman, Boris *et al.* (2019). *nomencl: a package to create a nomenclature.* TUG. Version 5.1. Readable with `texdoc nomencl`.

▷ Claudio Beccari
Emeritus Professor
of Politecnico di Torino
`claudio dot beccari at polito dot it`