LaTeX programming
https://www.latex-project.org/

Basic tutorial to LaTeX programming

Sébastien LE ROUX   sebastien.leroux@ipcms.unistra.fr

INSTITUT DE PHYSIQUE ET DE CHIMIE DES MATÉRIAUX DE STRASBOURG,
DÉPARTEMENT DES MATÉRIAUX ORGANIQUES,
23 RUE DU LOESS, BP43,
F-67034 STRASBOURG CEDEX 2, FRANCE

OCTOBER 19, 2020

Contents

LATEX ?

LATEX, shortcut for 'Lamport TEX', is a document markup language invented by Leslie Lamport in 1983. The purpose of LATEX is to simplify the utilization of the word processor TEX developed by Donald Knuth since 1977.

With typical word processors such as Microsoft Word and LibreOffice Writer, called **WYSIWYG** editors "**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et", one can immediately visualize the formatted text and the final shape of the document on the screen.

In LATEX the writer uses plain text (as opposed to formatted text), relying on markup tagging conventions to:

- define the general structure of a document (such as article, book, and letter).

- stylize text throughout a document (such as bold and italic).

- insert objects in the document (such as tables and figures).

- add citations and cross-referencing.

A TEX distribution such as TeX Live or MiKTeX is used to produce an output file (such as PS, PDF or DVI) suitable for printing or digital distribution.

The latest version of LATEX is called LATEX 2ε.

Many ideas in this HowTo where inspired by the very good book "LATEX par la pratique" (in French) by Christian Rolland [1] ...

... and many others by the LATEX Wikibooks: https://en.wikibooks.org/wiki/LaTeX

1.1   WYSIWYG vs. LATEX

LATEX being a programming language it is required to learn it, or at least a part of it, which scares most of the potential candidates to use it.

Using a WYSIWYG editor you can visualize the result of your work immediately. With LATEX few steps are required to obtain the final manuscript using the source file(s) in TEX language. The sequence of these steps is called compilation, and the compilation process can slightly change from one case to another.

1.2   Which LATEX ?

1.2.1   On-line: collaborative LATEX

Over the past few years, and like for many other sofware resources, the LATEX community has taken advantage of cloud computing. Several websites now offers complete and intuitive LATEX interfaces to edit and build your documents on line:

- https://fr.overleaf.com/

- https://www.sharelatex.com/

- https://papeeria.com/

This is a simple way to start and discover LATEX, and it is likely that these webistes offer one of the most interesting option, if not the best, to setup collaborative work using LATEX. Using such LATEX-in-the-cloud website would however require to have your research online with all the advantages and drawbacks that internet has to offer in such case.

1.2.2   Off-line: installing LATEX

1.2.2.1   Linux

To install TEX and LATEX 2ε on Linux you can simply use your favorite package manager (Yumex, Synaptic ...), see bellow for a list of packages.
You can also use the command line, remember that you need to be logged in a administrator.
To install the complete TeX Live distribution ($\sim$ 1.5Go):

- RedHat based Linux distributions:

```
user@localhost ~]$ sudo dnf install texlive-scheme-full
```

- Debian based Linux distributions:

```
user@localhost ~]$ sudo apt-get install texlive-full
```

To install the minimal Tex Live distribution on RedHat based Linux use:

```
user@localhost ~]$ sudo dnf install texlive
```

Afterwards the following commands ensure that you will install on your computer all the packages required to produce the examples proposed in this HowTo, many of which might already be installed by the previous command.

- Math and tables:

```
~]$ sudo dnf install texlive-amstex
~]$ sudo dnf install texlive-tabls texlive-multirow
```

- Images and graphics:

```
~]$ sudo dnf install texlive-graphviz texlive-xcolor
```

- Scientific editors:

```
~]$ sudo dnf install texlive-revtex texlive-achemso
~]$ sudo dnf install texlive-iopart tetex-elsevier
```

- Bibliography:

```
~]$ sudo dnf install texlive-bibtex
```

- Others:

```
~]$ sudo dnf install texlive-hyperref texlive-pslatex
~]$ sudo dnf install texlive-babel
```

1.2.2.2   Mac

You need to install the MacTeX distribution:

- Complete MacTeX package ($\sim$ 2Go): https://tug.org/mactex/mactex-download.html

- Smaller MacTeX distribution: https://tug.org/mactex/morepackages.html

1.2.2.3   Windows

The best choice is probably to install the MiKTeX distribution which is an up-to-date implementation of TeX, LaTeX and related programs for Windows:

- Internet installer: http://miktex.org/download

Alternatively you can install the TeX Live distribution:

- Internet installer: https://www.tug.org/texlive/acquire-netinstall.html

- Obtain TeX Live on DVD: http://https://www.tug.org/texlive/acquire-dvd.html

1.3 Before running LATEX

1.3.1 WYSIWYG-like tools for LATEX

To help you start you can install some tools that offer a WYSIWYG interface for LATEX:

- GNU TeXmacs: http://www.texmacs.org/

- BaKoMa TEX: http://www.bakoma-tex.com/

Using these softwares you can prepare your LATEX manuscript in a Microsoft Word or Libre-Office Writer kind of way, however they are limited and the source file they produce, if you generate some errors, and you will, is usually complicated to read.

1.3.2 LATEX editor

To start working you need to pick a text editor, if your OS is Microsoft Windows remember to use a "Unix Line Endings" editor. Indeed many Windows-based text editors add a particular symbol at the end of each line to signify that the line is breaking, and this symbol is messing with LATEX. Among the editors that can be used:

- The basic editors:

    - GNU Emacs
    - gVim

- The tools dedicated to LATEX:

    - Texmaker see figure [Fig. 1.1]
    - TEXnicCenter
    - TeXworks
    - TeXstudio

Hereafter I will use few examples from Texmaker, an opensource multiplateform (Linux, Mac, Windows) software that provides a user interface with spell-checking (see 5), help to many LATEX commands and a PDF reader.

1.3.3 How to install manually a package for LATEX

Add-on features for LATEX are known as packages. Dozens are pre-installed with LATEX and can be used immediately. They should all be stored in sub-directories of `texmf/tex/latex` named after each package. The directory name "`texmf`" stands for "TEX and METAFONT".
A package is a file or collection of files containing extra LATEX commands and programming which add new styling features or modify those already existing. There are two main file types delivered by packages:
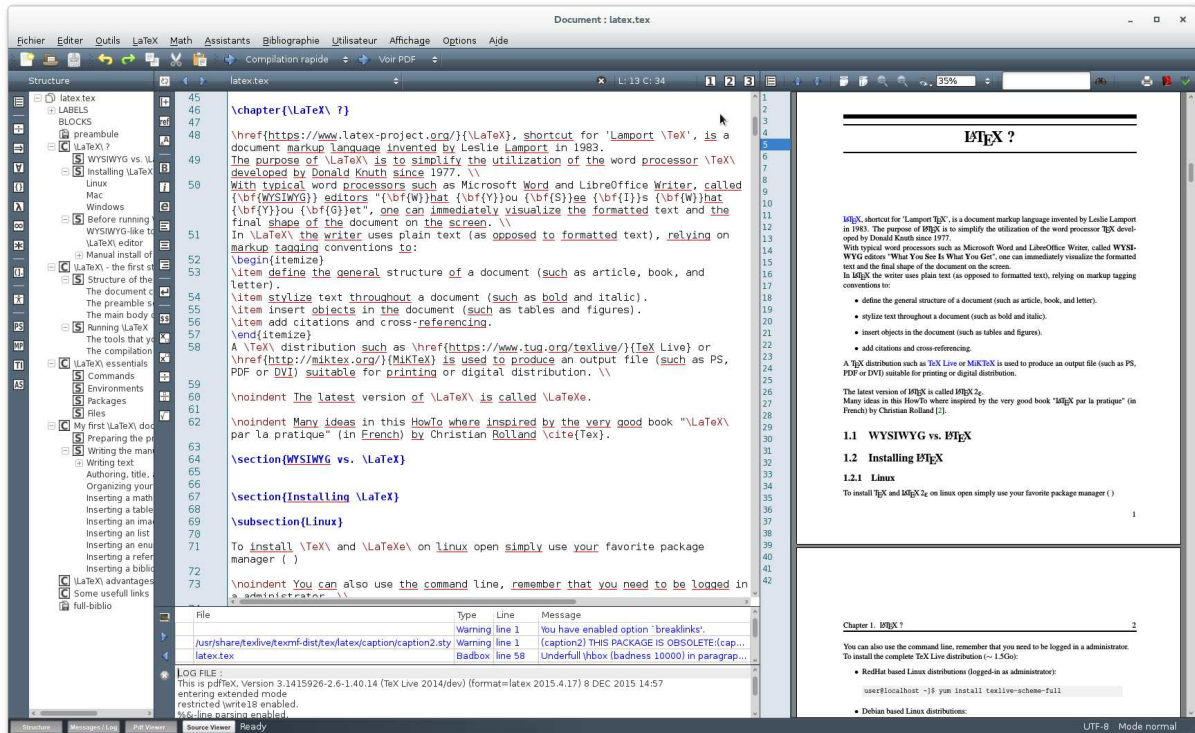
Figure 1.1   *The main window of the Texmaker editor*

- class files with `.cls` extension

- style files with `.sty` extension

When you try to typeset a document which requires a package which is not installed on your system, LATEX will warn you with an error message that it is missing [see 9]. Most LATEX installations come with a large set of pre-installed style packages, so you can use the package manager of the TEX distribution or the one on your system to manage them. But many more are available on the net. The main place to look for style packages on the Internet is Comprehensive TEX Archive Network, **CTAN**. Once you have identified a package you need that is not in your distribution, use the search engines of any CTAN server to find it. Download and follow this procedure to install it (after opening the archive):

1. Extract the files of the package: run LATEX on the `*.ins` file, ex:

   - For Linux / Mac:

     ```
     user@localhost ~]$ latex package.ins
     ```

   - For Windows open the `*.ins` using Texmaker and run a "Quick Build" [see 2.2.2]

2. Create the documentation: run LATEX twice on the `*.dtx` file, ex:

- For Linux / Mac:

```
user@localhost ~]$ latex package.dtx
user@localhost ~]$ latex package.dtx
```

- For Windows open the *.dtx using Texmaker and run a "Quick Build" [see 2.2.2]

3. Install the files:

   Packages installed manually should always be placed in your "local" directory, not in the directory tree containing all the pre-installed packages. This is done to prevent your new package to accidentally overwrite files in the main TEX directory.

The folder to look for should probably be called `texmf/` or `texmf-local/` and its location depends on your operating system:

- Linux: `$HOME/texmf/`

- Mac (MacTeX): `/Users/username/Library/texmf/`

- Windows (MikTeX): any folder registered as a user-managed texmf directory

The "right place" sometimes causes confusion, the "right place" for a LaTeX `.sty` file is a suitably-named sub-directory of `texmf/tex/latex/` "Suitably-named" means sensible, meaningful and probably short.

Often there is just a `.sty` file to move, but for complex packages there may be more, and they may belong in different locations. For example, new BibTeX packages or font packages will typically have several files to install. This is why it is a good idea to create a sub-directory for the package rather than dump the files into misc along with other unrelated stuff. In any case read the documentation to find out if there is a special or preferred location to move them to.

Table 1.1   *Where to put the files from the packages*

Type	Directory (in `texmf/` or `texmf-local/`)	Description
`.afm`	`fonts/afm/foundry/typeface`	Adobe Font Metrics for Type 1 fonts
`.bst`	`bibtex/bst/packagename`	BibTeX style
`.cls`	`tex/latex/base`	Document class file
`.dvi`	`doc`	package documentation
`.enc`	`fonts/enc`	Font encoding
`.fd`	`tex/latex/mfnfss`	Font Definition files for METAFONT fonts
`.fd`	`tex/latex/psnfss`	Font Definition files for PostScript Type 1 fonts
`.map`	`fonts/map`	Font mapping files
`.mf`	`fonts/source/public/typeface`	METAFONT outline
`.pdf`	`doc`	Package documentation
`.pfb`	`fonts/type1/foundry/typeface`	PostScript Type 1 outline
`.sty`	`tex/latex/packagename`	Style file: the normal package content
`.tex`	`doc`	TeX source for package documentation
`.tex`	`tex/plain/packagename`	Plain TeX macro files
`.tfm`	`fonts/tfm/foundry/typeface`	TeX Font Metrics for METAFONT and Type 1 fonts
`.ttf`	`fonts/truetype/foundry/typeface`	TrueType font
`.vf`	`fonts/vf/foundry/typeface`	TeX virtual fonts
`others`	`tex/latex/packagename`	Other types of file unless instructed otherwise

4. Update your index:

   Run your TeX indexer program to update the package database. This program comes with every version of TeX and has various names depending on the LATEX distribution you use:

   - Linux - TeX Live: `texhash`

   ```
   user@localhost ~]$ texhash
   ```

   - Mac - MacTeX does the job for you.
   - Windows - MiKTeX:
     - Windows XP through 7: Start menu $\rightarrow$ All Programs $\rightarrow$ MiKTeX $\rightarrow$ Settings
     - Windows 8: use the keyword Settings and choose the option of Settings with the MiKTeX logo.

   In Settings menu choose the first tab and click on Refresh FNDB-button, after that just verify by clicking 'OK'.

LATEX - the first step

2.1 Structure of the document

In the most simple case the structure of a LATEX document is as follow:

```
\documentclass{article}
```
$\implies$ the document class - [see 2.1.1]

```
% Some (important) information here
```
$\implies$ the preamble section - [see 2.1.2]

```
\begin{document}
```
$\implies$ the article starts here

```
Science is great, I love it !
```
the main body section - [see 2.1.3]

```
\end{document}
```
$\implies$ the article ends here

The LATEX file is a simple text file with the `*.tex` extension.

2.1.1 The document class

The document class is the first line of your LATEX file, it is a command that tells LATEX which type of document you are going to write, among the default choices provided you can find:

- letter $\implies$ to write a letter
- article $\implies$ to write an article
- book $\implies$ to write a book (like your thesis)
- beamer $\implies$ to create a presentation

On the document class declaration line you can also find additional options, or keywords, separated by a coma:

```
\documentclass[a4paper,12pt]{article}
```

 In this example the first keyword specify the paper layout, the second is used to change the default font size in the entire document. 3 different font sizes are available in the `book` and `article` classes: `10pt`, `11pt` or `12pt`, other are available using packages [see 3.3].
For more details about the different keywords or options, please refer the documentation of the corresponding document class.

Almost all scientific journals provide their own guidelines for LATEX publishing, including the document class you need to use, the trick is only, when needed, to download the class from the web site of the journal and to know how to install it [See 1.3.3]. For example the **A**merican **P**hsyical **S**ociety, **APS** (*Phys. Rev., Phys. Rev. Lett. ...*) and the **A**merican **I**nstitute of **P**hysics, **AIP** (*J. Chem. Phys., Appl. Phys. Lett. ...*) use the **revtex** class:

```
\documentclass[prb,twocolumn,showkeys,showpacs,english]{revtex4-2}
```

 IOP Publishing (*J. Phys.: Cond. Mat., Rep. Prog. Phys. ...*) uses the **iopart** class:

```
\documentclass[jpcm,twocolumn,english]{iopart}
```

 The **A**merican **C**hemical **S**ociety, **ACS** (*J. Am. Chem. Soc., Chem. Mater. ...*) uses the **achemso** class:

```
\documentclass[jacs,twocolumn,preprint,english]{achemso}
```

Elsevier journals (*J. Non-Cryst. Solids., Comp. Mat. Sci. ...*) use the **elsarticle** class:

```
\documentclass[jncs,twocolumn,preprint,english,12pt]{elsarticle}
```

 You must have noticed in these examples that a keyword is used to specify the journal it-self in the options of the document class. Using this keyword LATEX will reproduce the exact layout of the scientific journal you are targeting, from the title page to the style of the bibliography.

2.1.2   The preamble section

In this section of the text file you will tell LATEX about all the tools that you need to use:

- Packages = TEX or LATEX extensions that you need to use - [see 3.3]

- Commands = TEX or LATEX commands [see 3.1] that:

    – you might need to use in the preamble, these likely depend on the document class.
    – you might need to create and use in the preamble - [see 2.1.3]
    – you might need to create to use later in the main body of the manuscript - [see 2.1.3]

- Files that can be read by LATEX before processing the rest of the document - [see 3.4]

2.1.3   The main body of the manuscript

In this section you write your manuscript, your truth about science, and whatever comes to your mind and that can be published !
This manuscript section can contain:

- Basic text

- Commands = TEX or LATEX commands [see 3.1] to do many things like:

    – Insert a title $\Longrightarrow$ [see 4.2.2]

    – Organize your document $\Longrightarrow$ [see 4.2.3]

    – Insert a math equation $\Longrightarrow$ [see 4.2.4]

    – Insert a table $\Longrightarrow$ [see 4.2.5]

    – Insert an image $\Longrightarrow$ [see 4.2.6]

    – Insert a list (like this one) $\Longrightarrow$ [see 4.2.7]

    – Insert a web link $\Longrightarrow$ [see 4.2.9]

    – Insert a file $\Longrightarrow$ [see 3.4]

    – Insert a reference to one of the above $\Longrightarrow$ [see 4.2.8]

    – Insert a bibliographic reference $\Longrightarrow$ [see 4.2.10]

    – Insert a bibliography $\Longrightarrow$ [see 4.2.10]

2.2   Running LATEX

2.2.1   The tools that you will need

- A text editor

- Working distributions of TEX and LATEX 2ε

- TEX extensions that you might want to use (BibTeX, LaTeX2HTML ...)

2.2.2   The compilation process

For the time being let us consider that you already prepared a TEX file that contains your manuscript. This file has to be a basic text file with the extension `*.tex`, ex: `manuscript.tex`. To visualize the manuscript requires to build it using this source file that contains the LATEX commands and the text. This building process is called compilation, depending on your OS and the tool you are using to prepare your document this can be achieved using 2 methods:

1. Using the command line and the following sequence of commands, to get more familiar with the command line see [2]:

   ```
   user@localhost ~]$ latex manuscript

   ... many blabla

   user@localhost ~]$ latex manuscript

   ... many blabla
   ```

   You need to run the command `latex` twice, the second time is mandatory to cross-check and update references [see 4.2.8] and citations [see 4.2.10]. The `latex` command will produce at least 3 files:

   - `*.dvi` *device independent*, your manuscript.
   - `*.log` LATEX log journal with information about the last compilation.
   - `*.aux` LATEX auxiliary file.

   Afterwards if you want to produce a PostScript, *.ps, use:

   ```
   user@localhost ~]$ dvips manuscript
   ```

   And to build a PDF file, use:

   ```
   user@localhost ~]$ dvipdf manuscript
   ```

   `manuscript` is the name of your TEX file without extension, thus without the `*.tex`.

2. Using the graphical user interface of your LaTeX editor:
   In the Texmaker editor do the following:

   (a) Open the "Options → Configure Texmaker" dialog:
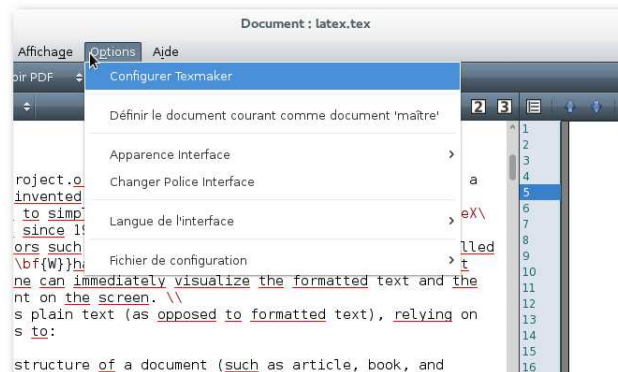
Figure 2.1   *Open the "Configure Texmaker" dialog box.*

   (b) Browse the "Quick Build" ("Compilation Rapide" in French) tab, and select the
       "LaTeX + dvips + ps2pdf + See PDF" option:
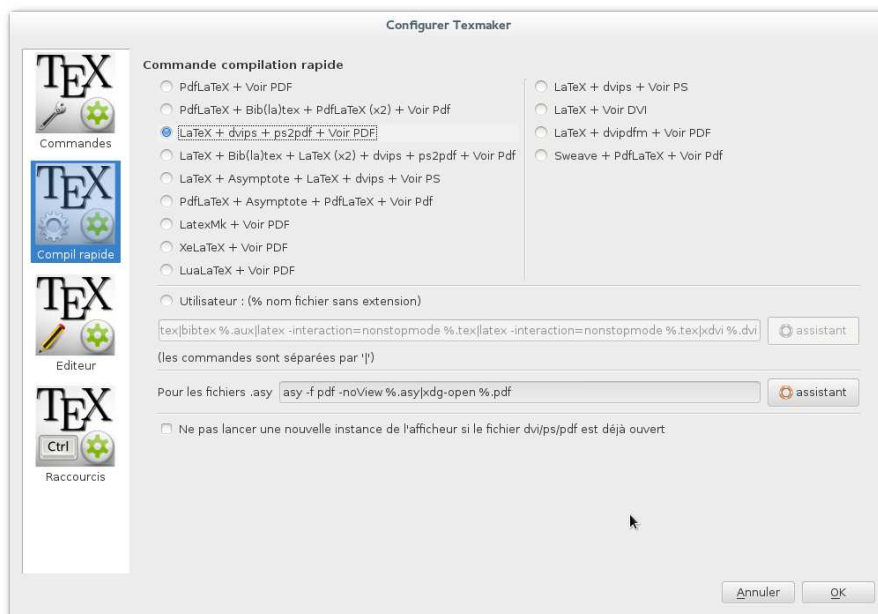
Figure 2.2   *The "Quick Build" tab in the Texmaker options dialog box. At this point you*
                  need to select the "LaTeX + dvips + ps2pdf + See PDF" option.

   (c) Use "Quick Build" button to prepare the PDF of your manuscript.

The compilation is a crucial step in the production of scientific document with LaTeX, [see 6,9] for more details.

LATEX essentials

3.1 Commands

Since LATEX is a programming language it is fundamental to be able to recognize a command, and you probably already noticed that without commands there is not much that you can do no matter how powerful LATEX is and how great is the thesis that you want to write.
In LATEX a command looks like:

```
\dosomething
```

You can also add an option to the command, placed in-between brackets, **{ }**, immediately after the command:

```
\dosomething{option}
```

And even more than one:

```
\dosomething{option 1}{option 2}...
```

Many, many commands have already been written and can be used to do many, many things, in the following parts of this HowTo I will shortly present some of the most basic, important, fundamental commands that you want to know to be able to write you precious PhD thesis or your next article using LATEX.
However before this, I think that it can be useful to know how to define your own command in LATEX:

```
\newcommand{\MyCommand}{What-it-does}
```

Here **\newcommand{}** is the LATEX command to define a command, the name of this new command will be **\MyCommand**, and it will perform the **{What-it-does}** operations, exam-

ple:

```
\newcommand{\napo}{[(P$_2$O$_5$)$_{1-x}$ (B$_2$O$_3$)$_x$]$_{0.65}$}
```

As a result, each time that LATEX will find the newly defined **\napo** command in the document, it will replace it by its content and print: $[(P_2O_5)_{1-x} (B_2O_3)_x]_{0.65}$.

You might also want to add options to your command:

```
\newcommand{\MyCommand}[1]{What-it-does-#1}
```

The number in red represents the number of options for the command, when defining the command the option is referred to using #1, example:

```
\newcommand{\green}[1]{\textcolor{green}{#1}}
```

The **\textcolor{col}{}** is the standard LATEX command to change the font color of the text between brackets, **{}**, to the color col. The **\green** command is simply a shortcut to change the text color to green, thus each time LATEX will find the **\green{**something is written here**}** command in the document, it will colored the text between brackets, **{}**, in green like this: something is written here.
Of course more than one option can be used when defining your command:

```
\newcommand{\MyCommand}[2]{It-does-that-#1-and-this-#2}
```

Finally you might need to redefine an existing command, you do this using the instruction **\renewcommand{}**, example:

```
\renewcommand{\figurename}{Fig.}
\renewcommand{\tablename}{Tab.}
```

Theses commands are used to define how the captions of inserted figures and tables start, the default values being respectively "Figure" for **\figurename** and "Table" for **\tablename**.

3.2 Environments

You should now be more familiar with the idea of command in LATEX many many commands exist, many of them being available without any particular requirement. However many others can only be used inside what is called an environment, ie. a part of the document where the standard behavior of LATEX is modified to do particular things. To do these things LATEX will give you access to a different set of commands not accessible otherwise.

To use an environment you need to open it with the **\begin** command, and when done with it close it using the **\end** command:

```
\begin{something}
I want do something special here !
\end{something}
```

Among the many environments some very very useful, if not essentials, can be used to insert equations [see 4.2.4], tables [see 4.2.5], images [see 4.2.6], lists and enumerations [see 4.2.7], a bibliography [see 4.2.10] ... and much more.

Always remember when you opened an environment, using the **\begin** command, to close it, using the **\end** command.

Many environments are used to insert floating objects, that can be moved in the document independently from the main text body, like equations, tables or images. LaTeX will handle floating objects to maintain the best possible presentation, however there may be, and will be times when you disagree. In this case additional options, keyword letters and symbol, to the **\begin** command can be used to specify the position:

```
\begin{something}[htbp!]
I want do something special here !
\end{something}
```

option	What it does
h	Put the object **here**, i.e. "*approximately*" the same point that in the source text.
t	Put the object at the **top** of the page.
b	Put the object at the **bottom** of the page.
p	Put the object on an isolated **page**.
!	Override internal LaTeX parameters to determine "*good*" positions (exact position).

Table 3.1   *Keyword letters and symbol to specify the position of floating objects.*

For example:

```
\begin{table}[!h]
I want do something special here !
\end{table}
```

Will insert a table, the options [!h] will ensure that it is inserted at the exact position it appears in the source text.

Like it is possible to define your own command(s), it is possible to define your own environ-
ment(s), however it is slightly more complicated. This is done using the **\newenvironment**
command:

```
\newenvironment{my_env}[Arg.][Def.Opt.]{% Def.\begin}{% Def.\end}
```

my_env is the name of the new environment, Arg. is the number of arguments that you want
to pass to the environment (similar to 3.1), Def.Opt. is an optional parameter that gives the
default value of the first argument, the two following parameters describe what to do when
opening the environment (**\begin** command) and when closing it (**\end** command).
An alternative method is to use a set of **\def** commands:

```
\def\my_env{% Here what to do at \begin{my_env}}
\def\endmy_env{% Here what to do at \end{my_env}}
```

The first **\def\my_env** command defines what to do when my_env is opened, the second
using **\def\endmy_env** defines what to do when the my_env.
The following examples can be used to define an abstract environment:

```
\newcommand{abstract}
  [1][ ]
  { \begin{center}
    \begin{minipage}{8cm}
    \begin{center}
    {\bf Abstract}
    \end{center}
    \par\small }
  {\end{minipage}
    \end{center} }
```

```
\def\abstract{
    \begin{center}
    \begin{minipage}{8cm}
    \begin{center}
    {\bf Abstract}
    \end{center}
    \par\small}

\def\endabstract{
    \end{minipage}
    \end{center}}
```

And it could be simply used like this example:

```
\begin{abstract}
This great work is about
to blow your mind ! is it ?
\end{abstract}
```

Abstract

This great work is about to blow your mind ! is it ?

The minipage environment allows to create a new text area. This area created using the
\begin{minipage} command can contain anything that a normal text page can contain (text,
images, tables ...).

3.3 Packages

As already mentioned packages are extensions that add functionalities to LaTeX, and there are many, many packages. If you want to use a package you need first to have this package installed [see 1.3.3] and then to tell LaTeX about it the preamble section using the **\usepackage** command:

```
\usepackage{mypackage}
```

With this command LaTeX will load the package mypackage, before continuing to process the rest of the .tex file.
Some packages do have options, that you can specify using keywords, separated by a coma, in between [ ] before the package name:

```
\usepackage[option1,option2,option3]{mypackage}
```

An comprehensive package list is given afterwards [see 4.1], it includes most of the packages you might need to prepare a scientific article or your thesis.

3.4 Files

When writing your significant contribution to science keep in mind that clarity is essential. Thus it can be cleaver to divide your LaTeX file in several pieces, for example information regarding each of the different chapter of your thesis could be in different files.
Two different commands exist to insert a file in a LaTeX document, the **\input** command and the **\include** command:

```latex
\documentclass{book}

% Preamble section

\input{preamble}                          ⟹   reading the file: preamble.tex

\includeonly{intro,chapter1}

\begin{document}

% Main body section

\include{intro}                           ⟹   reading the file: intro.tex

\include{chapter1}                        ⟹   reading the file: chapter1.tex

\include{chapter2}                        ⟹   reading the file: chapter2.tex

\include{conclu}                          ⟹   reading the file: conclu.tex

\end{document}
```

LATEX automatically adds the `.tex` extension to the file names given by the **\input** and **\include** commands.

Therefor in this example LATEX will look for the file `preamble.tex` in the preamble section, this file can contains **\usepackage**, commands and environments declaration(s).

Afterwards in the main body section, LATEX will read the files `intro.tex`, `chapter1.tex`, `chapter2.tex` and `conclu.tex` containing different the parts of the manuscript.

Only the main `.tex` file contains the **\documentclass**, **\begin{document}**, and **\end{document}** commands. The contents of the file which name is specified by the **\input** or the **\include** commands is simply inserted at the location of the command during the compilation process.

The **\include** command inserts the file in the document using a new page, while the **\input** command inserts the file without condition.

Also the **\include** command can be activated or deactivated by the **\includeonly** command. Therefore in the previous example only the files `into.tex` and `chapter1.tex` will be inserted in the main document during the compilation. The combination of **\include** the **\includeonly** command can be very useful when preparing large documents with extended chapters like your thesis.

My first LaTeX document

4.1 Preparing the preamble

Before starting to write your manuscript you need to ensure that all packages and commands that this work might requires are listed in the preamble of the main LaTeX file. You can define your own commands here [see 3.1], remember that doing so can considerably help to simply the writing of the manuscript. Finally you can ask latex to read file(s) to get extra information. An interesting example of the preamble section of your main LaTeX file is the following:

```latex
\documentclass{article}

\input{preamble}

\begin{document}
```

Here the preamble section consists of a single command, that tells LaTeX to read a file, this file called `preamble.tex` contains all the information regarding the packages and commands. One of the advantages of this method is to have a main file easier to read, since it now only contains your precious manuscript. Also if you decide to keep working using LaTeX you could recall the same `preamble.tex` file.

A comprehensive example of the preamble section is given afterwards, and most of the packages you might need to write your piece of science and print your mark in history are listed hereafter, you just need to adapt it to your personal use:

```
% To have an input font encoding that support special characters
\usepackage[utf8]{inputenc}
% To have an output font encoding that support special characters
\usepackage[T1]{fontenc}
% To define the language of the document
\usepackage[english]{babel}
% Much better font for the PDF file
\usepackage{pslatex}
% Hyperlinks for the PDF file
\usepackage[pdfauthor     = {Sébastien Le Roux},
            pdftitle      = {LaTeX tutorial},
            pdfsubject    = {LaTeX tutorial},
            pdfkeywords   = {LaTeX Users Manual Guide Basics Help},
            pdfcreator    = {LaTeX+DviPDF},
            pdfproducer   = {LaTeX+DviPDF},
            pdfstartview  = FitV,  % Adjust the document to the window at startup
            dvips         = true,  % Use hyperref with dvips
            colorlinks    = true,  % Colored hypertext links
            plainpages    = false, % Do page number anchors as plain arabic
            pagebackref   = true,  % Allows to add links in the bibliography ...
            backref       = page,  % .. that points toward the appropriate pages
            hyperindex    = true,  % Add hypertext links in the appendix
            linktocpage   = true,  % Links on the page numbers and not the text
            breaklinks    = true,  % To write the long hyperlinks on more than one line
            urlcolor      = blue,  % Color for external hyperlinks
            linkcolor     = red,   % Color for internal hyperlinks
            bookmarks     = true,  % To create the section marks pour Acrobat Reader
            bookmarksopen = false  % Is the document tree entirely opened at startup
]{hyperref}

% Insert images
\usepackage{graphicx}
% Colors
\usepackage{xcolor}
% Enumerations
\usepackage{enumerate}
% Mathematics
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{amscd}
\usepackage{theorem}
% Tables
\usepackage{hhline}
\usepackage{multirow}
\usepackage{tabls}
% To use landscape pages in a portrait document
\usepackage{lscape}
% Spaces between lines
\usepackage{setspace}
%\onehalfspacing
%\doublespacing
%\setstretch{3}
```

4.2   Writing the manuscript

From now on every command, environment, text is going to be written in the main body of the document, thus after the line **\begin{**document**}**.

4.2.1   Writing text

4.2.1.1   Special characters

You are now ready to prepare your first manuscript, before starting you need to know that LaTeX considers some particular characters as commands. If you do not want LaTeX to shout at you during the compilation process [see. 9] you should be familiar with these 10 characters:

$$\backslash \quad \# \quad \% \quad \sim \quad \{ \quad \} \quad \$ \quad \_ \quad \hat{} $$

- \ is the beginning of a command - [see 3.1]

- # is used when creating commands - [see 3.1]

- % starts a comments, everything that follow on the same line is not interpreted, read or compiled by LaTeX

- ~ creates a non-breaking space

- { and } highlight the beginning and the end of a command or an environment

- $, _ and ^ - [see 4.2.4]

- & - [see 4.2.5]

4.2.1.2   Few basic, yet very useful, commands

\LaTeX	to print LATEX
\TeX	to print TEX
\today	to print the date, today is: October 19, 2020
****	to force LATEX to jump to the next line.
****[?cm]	to force LATEX to jump to the next line, with a space of ? cm
\vspace[?cm]	to force LATEX to create a vertical space of ? cm
\hspace[?cm]	to force LATEX to create a horizontal space of ? cm
\newpage	to force LATEX to start a new page.
\clearpage	to force LATEX to start a new page ...
	... and to print all pending objects (tables, figures ...) from the stack.
\indent	the first line of the upcoming paragraph should be shifted.
\noindent	the first line of the upcoming paragraph should not be shifted.

4.2.1.3   Font size

You can change the font size, for the entire document [see 2.1.1], or selected parts of it.
Here are the commands to change the font size for a part of the document:

{\tiny Bigger**}**	Bigger
{\scriptsize Bigger**}**	Bigger
{\footnotesize Bigger**}**	Bigger
{\small Bigger**}**	Bigger
{\normalsize Bigger**}**	Bigger
{\large Bigger**}**	Bigger
{\Large Bigger**}**	Bigger
{\LARGE Bigger**}**	Bigger
{\huge Bigger**}**	Bigger
{\Huge Bigger**}**	Bigger

4.2.1.4   Font style

To change the font style use:

`\textrm{Roman}`	`{\rm Roman}`	(default)
`\textit{Italic}`	`{\it Italic}`	*Italic*
`\textsl{Sloping}`	`{\sl Sloping}`	*Sloping*
`\textbf{Bold}`	`{\bf Bold}`	**Bold**
`\textsc{Small capitals}`	`{\sc Small capitals}`	SMALL CAPITALS
`\emph{Emphasis}`	`{\em Emphasis}`	*Emphasis* *
`\texttt{Typewriter}`		`Typewriter`
`\uline{Underline}`		Underlined
`\uuline{Underline twice}`		Underlined twice
`\sout{Crossed}`		~~Crossed~~
`\xout{Shaded}`		////////

* the result depends on the font environment the command is used in:

- I write in Roman mode and I want to *emphasis this passage*

- *I write in Italic mode and I want to* emphasis this passage

If needed the font style can be ajdusted by embeding style commands into each others:

`{\it {\bf ItalicBold} }`	***ItalicBold***
`{\bf {\sc BoldSmallcapitals} }`	**BOLDSMALLCAPITALS**

4.2.1.5   Font color

The command to change the font color is:

```
\textcolor{col}{My colored text here !}
```

`col` is a color name, accessible using the `xcolor` package, many standard color names are available (red, green, blue ...) if the color you want is not available you can define it using:

```
\definecolor{my_green}{rgb}{0.0,0.6,0}
```

The last arguments of the **\definecolor** commands are 3 numbers between 0.0 and 1.0 that represent the color **RGB** code, with the **R**ed, **G**reen and **B**lue values.
The color must be defined before using it, therefore the **\definecolor** command must be used before the **\textcolor**. We suggest to define all the colors you need in the preamble section.

4.2.2   Title, authoring and abstract

It can be useful to inform your reader about what they are going to read, depending on the document class (book, article, revtex4-2, achemso ...) the instructions can change slightly for details give a look to the help pages of the corresponding class. This example can be used for most document classes:

```latex
% Give the title of your thesis / article
\title{My revolutionary science project}

% Name the authors
\author{M. Me, Y. You and H. He}

% You can even provide an abstract
\begin{abstract}
This great work is about to blow your mind !
\end{abstract}

% Now create the title page using all of the above
\maketitle
```

Detail examples for each editor related document classes are provided in [Chap. 7].

4.2.3   Organizing your document

If you want people to read and understand your work, it is important to organize it properly. That means divide it in parts, chapters, sections or other pieces that help to clarify its content for the reader. All these commands (see Tab. 4.1) have a similar structure and usage, hereafter we will use the **\section** as an example of any of the commands in table [Tab. 4.1].

```latex
\section[Section shortname]{Section title}
```

The first option of the command, the [ ], is not mandatory and can be used to modify the name of the section that will appear in the **T**able **O**f Contents, **TOC**, if any. The default name of the section in the TOC is given by the second option of the command, its real title in the document.

```latex
\section{My great section}

\subsection{My great subsection}
```

$\implies$

1   My great section

1.1   My great subsection

Table 4.1   *LATEX commands to organize your document.*

Command	Document class
\part	book, article*
\chapter	book
\section	book, article*
\subsection $\Longrightarrow$	book, article*
\subsubsection	book, article*
\paragraph	book, article*
\subparagraph	book, article*

* Including the revtex4-2, iopart, elsarticle and achemso classes.

The numbering of the chapters, parts, sections, etc, will be done automatically by LATEX, thus there is no need for the user to specify any number.

If you do not want the section to be numbered use a * symbol:

```
\section*{My great section}
```
$\implies$

My great section

```
\subsection*{My great subsection}
```
My great subsection

Note that the section(s) will still appears in the TOC, but without number(s).

One more command that you might need to organize your document is **\appendix** that indicates the end of the main part of the manuscript and the start of the appendix. In the appendix you use all the commands presented in table [Tab. 4.1] however the numbering will not be done with numbers anymore but with letters:

```
\chapter{My final chapter}

% Final chapter body

\section{My first section}

% First section body

\section{My second section}

% Second section body

\chapter*{Conclusion}

% The conclusion

\appendix

\chapter{Additional data}
```
$\implies$

Chapter 5
My final chapter

1   My first section

2   My second section

Conclusion

Appendix A
Additional data

In some classes, like book, it is possible to create a TOC, a list of figures and a list of tables:

```
% Create the table of content
\tableofcontent
% Add a list of figures (= list of all \begin{figure} and caption)
\listoffigures
% And a list of tables (= list of all \begin{table} and caption)
\listoftables
```

4.2.4   Inserting a math equation

They are two main possibilities to insert a math equation, or to use what is called the mathematics mode:

- the standard, in-line, math mode: **$**Math expression here**$**

- the equation math mode: **\begin{**equation**}**Math expression here**\end{**equation**}**

The math modes present specific properties:

- Text letters (a,A $\rightarrow$ z,Z) are written in italic ($a, A \rightarrow z, Z$), however numbers, symbols and punctuation signs are written in standard roman characters.

- Superscript writing using the ^ character and subscript writing using the _ character:

```
$6.022\times10^{23}$ \\
CH$_3$CH$_2$OH \\
NH$_3^+$
```
$\Longrightarrow$
6.022×10^{23}
CH_3CH_2OH
NH_3^+

  Note that if no brackets, **{ }**, are used, then only the first letter/symbol/number following the ^ and the _ characters will be affected and upper-lower-scripted.

- Many, many, many symbols are accessible using commands, such as:

  - Greek letters: **\alpha   \beta   \gamma   \Gamma   \delta   \Delta**

    α β γ Γ δ Δ

  - Math operators: **\sum   \prod   \frac{**1**}{**2**}   \sqrt[**x**]{**y**}   \int**

    $\sum$ $\prod$ $\frac{1}{2}$ $\sqrt[x]{y}$ $\int$

  - Math accents: **\vec{**v**}   \dot{**a**}   \bar{**h**}   \widehat{**abc**}**

    $\vec{v}$ $\dot{a}$ $\bar{h}$ $\widehat{abc}$

  - Symbols: **\hbar   \infty   \times   \div   \pm**

    $\hbar$ ∞ $\times$ $\div$ $\pm$

Using the first method, using $ $ symbols, also called 'in-line' equation it is possible to insert maths in the text, for example the instructions:

```
$\sum_{i=1}^{N} \frac{x-1}{\sqrt{\delta}} = \infty$
```

will be rendered like this in a text line, $\sum_{i=1}^{N} \frac{x-1}{\sqrt{\delta}} = \infty$, allowing to continue the discussion.

The other method using an `equation` environment adds a number to your equation, so that you can refer to this equation somewhere else in the document [see 4.2.8]:

```
\begin{equation}
\sum_{i=1}^{N} \frac{x-1}{\sqrt{\delta}} = \infty
\end{equation}
```

It will also be rendered separately from the text:

$$\sum_{i=1}^{N} \frac{x-1}{\sqrt{\delta}} = \infty \tag{4.1}$$

For other examples [see 7].

4.2.5   Inserting a table

To create a table it is required to use the `table` and the `tabular` environments, the first allows to insert a caption and make reference(s) to this table elsewhere in the document [see 4.2.8], while the second allows to organize elements in columns and rows:

```
\begin{table}
\begin{tabular}{l|cr||}
  & $\alpha$ & $\beta$ \\
\hline
\hline
a &  2.0 & $1\times10^{-4}$ \\
b & -3.0 & $\sqrt5$ \\
c &  1.0 & $1\times10^2$ \\
\hline
\end{tabular}
\caption{My interesting table}
\end{table}
```

	α	β
a	2.0	1×10^{-4}
b	−3.0	$\sqrt{5}$
c	1.0	1×10^2

Table 4.2   *My interesting table*

The options of the **\begin{**tabular**}** command define the shape of the table, in the previous example: **{**l|cr||**}**. The number of column(s) in the table is given by the number of letters, here 3 letters hence the table will have 3 columns. The letter itself defines how the text is centered in the corresponding column: l=left, c=center or r=right. The pipe symbol, **{**|**}** is used to draw separator(s) (by default straight lines) between columns.

In the table it-self, the **\hline** is used to draw separator(s) between rows. On the same row, columns are separated using the **&** symbol, the line is terminated using the \\ symbol.

In a multicolumn document it is possible to use the **\begin{**table***}** and **\end{**table***}** commands to produce a table that use the entire width of the page instead of the default width of a single column.

For other examples [see 7].

4.2.6   Inserting an image

To insert an image it is required to use the figure environment, to insert a caption and make reference(s) to this figure elsewhere in the document [see 4.2.8], and the **\includegraphics** command provided by **\usepackage{**graphicx**}** added to the preamble section [see 4.1]. The graphicx package allows to import and use most type of images, however it is **strongly recommended** to use **P**ost**S**cript, **.ps**, or preferably **En**capsulated **P**ost**S**cript, **.eps**, images with LATEX.

```
\begin{figure}
\includegraphics{image.eps}
\caption{My interesting figure}
\end{figure}
```

Figure 4.1   *My interesting figure*

The `\includegraphics` command accepts many optional keywords like:

- `keepaspectratio=true % or false`

- `scale=0.75`

- `width=5cm`

- `height=8cm`

- `angle=90`

- `draft=false`, this is the default value, if you set this option on `true` the image is not inserted, instead you will see a blank space of the same size, this could be very useful when preparing a large manuscript with many images, like your thesis.

- `natwidth=640`, used to solve bounding box errors [see 9]

- `natheight=480`, used to solve bounding box errors [see 9]

```
\includegraphics[width=10cm,keepaspectratio=true,draft=true]{image.eps}
```

In a multicolumn document it is possible to use the `\begin{figure*}` and `\end{figure*}` commands to produce a figure that uses the entire width of the page instead of the default width of a single column.

For other examples [see 7].

4.2.7   Inserting a list / enumeration

You can insert a list using the `itemize` environment and the **\item** command:

```
\begin{itemize}
\item Element 1
\item Element 2
\begin{itemize}
\item Element 3
\item Element 4
\begin{itemize}
\item Element 5
\item Element 6
\end{itemize}
\end{itemize}
\item Element 7
\end{itemize}
```

$\Longrightarrow$

- Element 1

- Element 2

  - Element 3

  - Element 4

    * Element 5
    * Element 6

- Element 7

For an enumeration use the `enumerate` environment and the **\item** command:

```
\begin{enumerate}
\item Element 1
\item Element 2
\begin{enumerate}
\item Element 3
\item Element 4
\begin{enumerate}
\item Element 5
\item Element 6
\end{enumerate}
\end{enumerate}
\item Element 7
\end{enumerate}
```

$\Longrightarrow$

1. Element 1

2. Element 2

   (a) Element 3
   (b) Element 4

       i. Element 5
       ii. Element 6

3. Element 7

4.2.8   Inserting a reference

To make a reference to an element can be achieved using the two commands **\label** and **\ref**. The first one is used to create a keyword associated with an object, the second one is used to make reference to this keyword and therefore to the object it-self:

```
\section{My third section}
\label{sec-3}

I like what's in figure \ref{pinte}

\section{My fourth section}




\begin{figure}
\includegraphics{image.eps}
\caption{\label{pinte}My interesting figure}
\end{figure}




As already said in section \ref{sec-3}

I like what's in figure \ref{pinte}
```

3   My third section

I like what's in figure 4.2

4   My fourth section

Figure 4.2   *My interesting figure*

As already said in section **3**

I like what's in figure 4.2

There is no need to number anything, LATEX does it for you. You can add a label using the **\label** command to pretty much any object in the document, including all the objects and environments presented previously in this HowTo. There is only one simple rule to respect for the floating objects, like the tables or the figures, and that is to put the **\label** command inside the **\caption** command. Finally remember that each label, in other word each KEYWORD, used in the **\label**{KEYWORD} command, must be unique.

4.2.9   Inserting web links and emails

To insert an hypertext link or an email address use the **\href** command. The **\href** command takes 2 arguments, the first one is the web link or the email address, the second is the text to be displayed in the final document, ex:

```
Link:\href{https://www.latex-project.org/}{The \LaTeX\ project web page}
Mail:\href{mailto:sebastien.leroux@icpms.unistra.fr}{Sébastien Le Roux}
```

 Will be be rendered as:
Link:The LATEX project web page
Mail:Sébastien Le Roux

4.2.10   Inserting a bibliography

There are 2 methods to create references and the corresponding bibliography in your TEX document, both require references to be inserted using the **\cite** command:

```
The work in this article is good, really good,
it is even better than some previous work in Ref.\cite{old_work_a},
it will clearly be more cited that Ref.\cite{old_work_b}
```

To the keyword used in the **\cite** command corresponds an entry in the bibliography that you need to create. You have 2 methods to achieve this goal:

1. Use the thebibliography environment, and create an entry for the reference using the **\bibitem** command:

   ```
   \begin{thebibliography}
   \bibitem{old_work_a}
   D. umb and I. diot,
   J. Non-Smart. Sci. \textbf{00}, 0000-0007 (2015).
   \bibitem{old_work_b}
   R. Car and M. Parrinello,
   Phys. Rev. Lett. \textbf{55}, 2471-2474 (1985).
   \end{thebibliography}
   ```

    You need to create all the entries in the bibliography your-self, eventhough it is not complicated it has to be done very carefully. You define the style of the citations yourself using LATEX commands, ie. the author name format, the order the authors list, journal, year, appear in ... Be careful to create the entries in the order you want them to appear in the bibliography section. To have the citations in the order they appear in the article/thesis can be tricky, in particular in you keep modifying the source file.

2. Use BibTeX: then you let LATEX, BibTeX and the scientific editors do the job for you. The only thing that you need then in your TEX file is to use the 2 following commands:

```
\bibliographystyle{unsrt}
\bibliography{biblio}
```

The first command **\bibliographystyle** set the style of the bibliography using a keyword, in this example unsrt, among others like: abbrv, alpha, plain.

Table 4.3 *The default BibTeX styles*

Keyword	Author name format	Reference Format	Sorting
abbrv	H. J. Simpson	Num.	by author
alpha	Homer Jay Simpson	Sim95	by author
plain	Homer Jay Simpson	Num.	by author
unsrt	Homer Jay Simpson	Num.	as referenced

Table [Tab. 4.3] presents few differences between the default BibTeX styles. Many others bibliography styles are available, in particular using the natbib package.

The **\bibliography** command tells LATEX to import the file biblio.bib (the .bib extension being added automatically) using BibTeX to create the bibliography. This so-called BibTeX file contains all the elements of the bibliography, in the BibTeX format with entries that looks like:

```
@Article{old_work_b,
    author = "Car, R. and Parrinello, M.",
    journal = "Phys. Rev. Lett.",
    volume = 55,
    number = 22,
    pages = "2471-2474",
    numpages = 3,
    year = 1985,
    publisher = "American Physical Society"
}
```

The first line starting with an @ gives the document type, here an article (among others like: book, phdthesis, proceedings, conference ...) and the associated keyword. The next lines are the fields to describe the bibliographic reference.

The software Mendeley provides a very nice graphical interface to help you manage your BibTeX file.

Why is BibTeX so great:

- Most of the scientific editors provide the BibTeX entries for their publications, which means that you only need to browse their web sites, find the page of the appropriate article, and copy-paste the entry in your own BibTeX file, then simply use the corresponding keyword in your manuscript.

- You can use the same BibTeX file for all your works with LaTeX, this means that you just need to do prepare an entry the first time you need the reference, then simply recall the appropriate keyword.

- No matter the number of entries in the BibTeX file LaTeX and BibTeX will only use the ones cited in your article and ignore the others.

- No need to take care of the order of the entries in the BibTeX file, the citations in your bibliography will be sorted automatically.

To use BibTeX the compilation sequence changes, and you need to enter the following sequence of commands:

```
user@localhost ~]$ latex manuscript
...
user@localhost ~]$ bibtex manuscript
...
user@localhost ~]$ latex manuscript
...
user@localhost ~]$ latex manuscript
...
```

In Texmaker you need to change the "Quick Build" options to "LaTeX + Bib(la)Tex + LaTeX (x2) + dvips + ps2pdf + See PDF", see figure [Fig. 4.3].
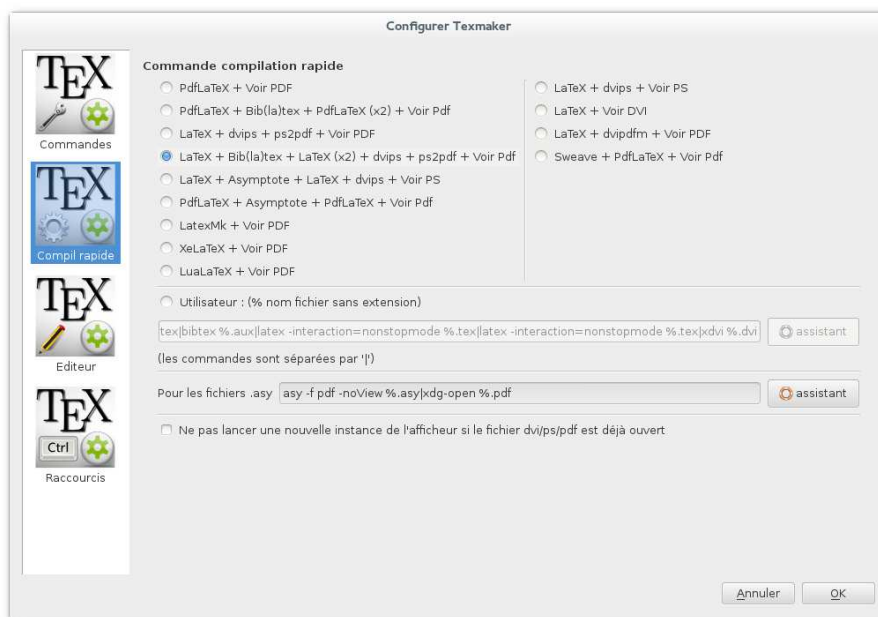
Figure 4.3   *The "Quick Build" tab in the Texmaker options dialog box for BibTeX. At this point you need to select the "LaTeX + Bib(la)Tex + LaTeX (x2) + dvips + ps2pdf + See PDF" option.*

Spell checking

5.1 Using the command line

You can check the spelling of your LATEX manuscript using the `aspell` command:

```
user@localhost ~]$ aspell -c -t --lang=en_EN manuscript.tex
```

The options are:

- `-c` : to check spelling.

- `-t` : the document to check is a TEX file, all TEX and LATEX commands will be ignored.

- `-lang=en_EN` : the language of the document (in this case English), among the dictionaries installed on your OS. For Linux and MacOS available dictionaries are listed in:

    - `/usr/share/myspell/`
    - `/usr/share/locale/`

Following this command you will enter an interactive mode to check the content of your TEX file, simply follow the instructions to correct it if necessary.

You could also used a BASH script, thus providing that the following script, named `check`, is installed somewhere in your PATH (see [2] for details):

```bash
#!/bin/bash

TFILE=$1
aspell -c -t --lang=en_EN $TFILE".tex"
```

You could spell check your LATEX document simply using:

```
user@localhost ~]$ check manuscript
```

5.2   Using Texmaker

1. Open the "`Options → Configure Texmaker`" dialog:
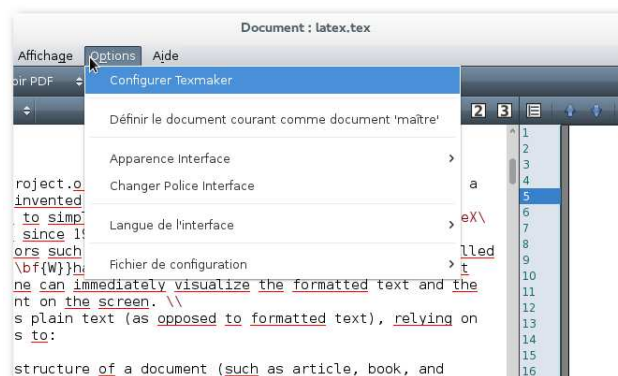
Figure 5.1   *Open the "Configure Texmaker" dialog box.*

2. Browse the "`Editor`" ("`Editeur`" in French) tab, and check that the appropriate dictionary is being used:
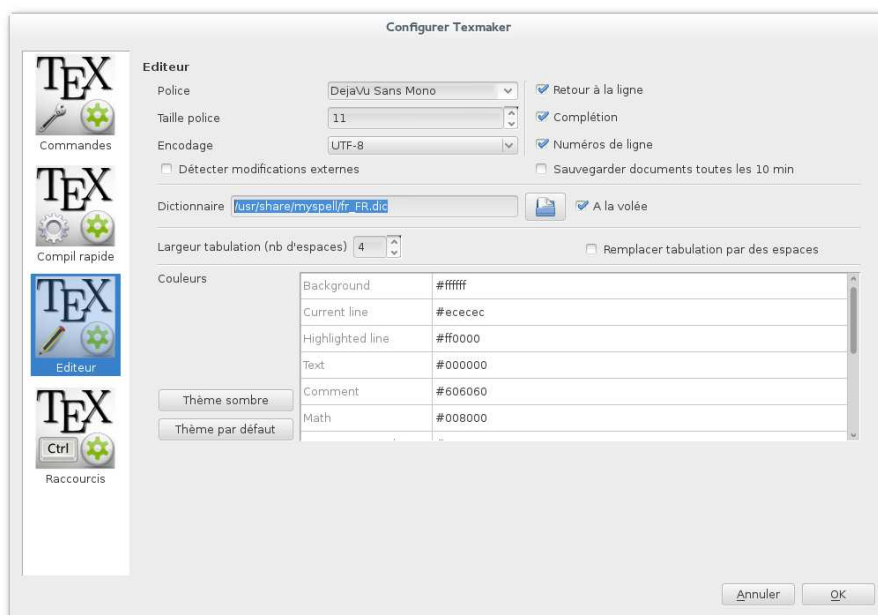
Figure 5.2   *The "Editor" tab in the Texmaker options dialog box. At this point you need to select the appropriate dictionary.*

3. Use "Check Spelling" button in the file menu (or the Ctrl+Maj+F7 keyboard shortcut) to check your manuscript:
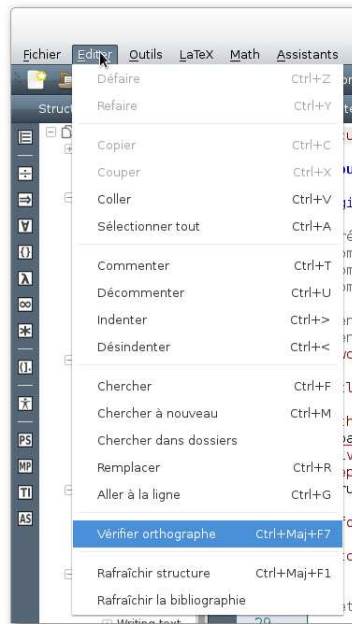
Figure 5.3   *Spell checking in Texmaker.*

For Windows dictionnaries are installed with Texmaker and can be found in:

- `C:\Program Files (x86)\Texmaker\`

LATEX advantages and drawbacks

Drawbacks

You will need to learn few things about programming

It is obvious that the first thing about LATEX is that you need to learn a bit, to say the least, about programming. If your are not willing to spend some time learning how things are behind the curtains, well forget about LATEX.

You will need to be patient

- Because you will not see immediately the result of your work.
  Most people spend a lot of time going back and forth between the source file and the PDF file. That is the wrong way to work with LATEX (and that is what you will start to do anyway): as soon as you inserted a few lines, or commands, you want to know how it looks like, and you build the PDF file.
  That works for small documents, forget about it for *real* ones, like your thesis or an article, in particular if you insert graphics, and you will. The more images you include, the longer it takes to build the PDF file, think about it when you will prepare your thesis (and also think about the `draft` option for the **\includegraphics** command).

- Because you will waste time trying to find that f*****g command your are looking for.
  That is the way it is, I wish I could tell you that you will be fluent in LATEX by the end of the day, and the commands will come to you naturally. Nope. As everybody, and even after years of experience, you will still waste time to look for a particular command to do that particular thing you really need to do.

- Because when you will find that f*****g command it will not work.
  And you will need to take time to install the package that contains this precious command that you really really need.

- Because the compilation will fail, many times - see bellow.

You will need to be organized

As you are a scientist, or about to become one, this should not be too much of a problem.
Why is it so important to be well organized:

- To avoid to waste too much time debugging errors during the compilation process.
  The compilation will fail, many times, because of silly mistakes like:

  - Forget to use an **\end** command to close a **\begin** command.
  - To miss or have an extra symbol: {, [, (
  - To miss or have an extra symbol: },],)
  - To miss or have an extra symbol: $
  - To miss or have an extra symbol: &

  Trying to find where the error comes from, will be time consuming, and as much as
  possible you want to avoid that.

- To keep your working directory clean enough to know what is going on.
  Indeed after the compilation process of your main `*.tex` file the directory will be filled
  with many files, created by LATEX and potentially BibTeX. It will also contains images,
  probably a significant number for your thesis. Think about creating sub-directory(ies) to
  sort and organize everything, it will be easier to find the information you might look for
  afterwards.

Advantages

If you are not too scared already, then hope for you there is young Padawan.
Your ally is the force in LATEX and a powerful ally it is !
Now is the time to learn why the light side of the force is the way to go.

May the force in LATEX be with you !

It looks great !

You will see by yourself soon enough that there is no way to match the quality of a document
produced using LATEX.

No more journal layout problems !

You just need to use the proper keyword, or option, usually in the **\documentclass**, some-
times in the preamble section, to have the job perfectly done by LATEX.

Cross-referencing and bibliography made easy !

No numbering to be done, simply use a different keyword for each object and LaTeX takes care of the rest: numbering, hypertext links, etc.
Using BibTeX you only have to do the job once, the first time that you add an entry in the `*.bib` file. After that simply re-use the appropriate keyword to cite the corresponding publication. This means that this BibTeX file can be shared between all your LaTeX projects.

You will save time, a lot !

Yes you will. Think about the time you would need with a WYSIWYG editor to:

- Prepare the layout of the document

- Organize properly the document

- Organize the cross referencing

- Organize the bibliography

Hopefully by now you understand that with LaTeX this can become almost automatic. Consequently you can save a lot of time to do something else.

Examples

7.1 Title and authors

7.1.1 RevTex - "revtex4-2"

```
\documentclass[prb,twocolumn,superscriptaddress]{revtex4-2}
% Use the "superscriptaddress" in the documentclass options.

\begin{document}

\title{My amazing work !}

\author{M. Me}
\affiliation{My great institution - France}
\author{Y. You}
\affiliation{My great institution - France}
\author{H. He}
\affiliation{The third author institution - USA}

\begin{abstract}
This great work is about to blow your mind !
\end{abstract}

\maketitle

% At this point the article start

\end{document}
```

7.1.2 American Chemical Society "achemso"

```latex
\documentclass[jacs,twocolumn]{achemso}

\title{My amazing work !}

\author{M. Me}
\affiliation{My great institution - France}
\author{Y. You}
\affiliation{My great institution - France}
\author{H. He}
\affiliation{The third author institution - USA}

\begin{document}

\begin{abstract}
This great work is about to blow your mind !
\end{abstract}

\maketitle

% At this point the article start

\end{document}
```

7.1.3 Elsevier "elsarticle"

```latex
\documentclass[jncs,twocolumn]{elsarticle}

\begin{document}

\title{My amazing work !}

\author[1]{M. Me}
\author[1]{Y. You}
\author[2]{H. He}
\address[1]{My great institution - France}
\address[2]{The third author institution - USA}

\begin{abstract}
This great work is about to blow your mind !
\end{abstract}

\maketitle

% At this point the article start

\end{document}
```

7.1.4 Institute Of Physics "iopart"

```
\documentclass[jpcm,twocolumn]{iopart}

\begin{document}

\title{My amazing work !}

\author{M. Me$^1$}
\author{Y. You$^1$}
\author{H. He$^2$}
\address{(1) My great institution - France}
\address{(2) The third author institution - USA}

\begin{abstract}
This great work is about to blow your mind !
\end{abstract}

\maketitle

% At this point the article start

\end{document}
```
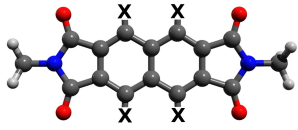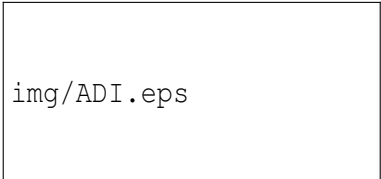
7.2   Tables and figures

```
\begin{table}[!h]
\caption{Label and description of the e- acceptor diimide cores studied.}
\vspace{0.3cm}
\begin{tabular}{rlcllc}
Id & Label & $N$ & Formula & \multicolumn{2}{c}{Diimide familly + 3D} \\
\hline
\\
 & & & & & \multirow{4}{5cm}
{\includegraphics[width=0.9\linewidth,keepaspectratio=true,draft=false]
{img/NDI.eps}} \\
 1 & NDI       & 32 & C$_16$H$_10$O$_4$N$_2$  &
\multirow{2}{4cm}{\bigg}Naphthalene} \\
 2 & NDI\_F4     & 38 & C$_{16}$H$_6$O$_4$N$_2$F$_4$ \\
\\
\\
 3 & ADI       & 38 & C$_20$H$_12$O$_4$N$_2$ &
\multirow{4}{4cm}{\Bigg}Anthracene} \\
\multirow{4}{5cm}
{\includegraphics[width=1.0\linewidth,keepaspectratio=true,draft=true]
{img/ADI.eps}} \\
 4 & ADI\_F2    & 38 & C$_{20}$H$_{10}$O$_4$N$_2$F$_2$ \\
 5 & ADI\_CN2   & 40 & C$_{22}$H$_{10}$O$_4$N$_4$ \\
 6 & ADI\_Br2   & 38 & C$_{20}$H$_{10}$O$_4$N$_2$Br$_2$ \\
\end{tabular}
\end{table}
```

The table above provides and example of the usage for the following commands:

- **\multicolumn**

  usage: **\multicolumn{**#NUM**}{**#LAYOUT**}{**Data to display**}}**
  with: #NUM= number of column, #LAYOUT = l=left, c=center, r=right

- **\multirow** (requires the multirow package)

  usage: **\multirow{**#NUM**}{**#WIDTH**}{**Data to display**}}**
  with: #NUM= number of row, #WIDTH = width of the text

- **\includegraphics**, comparison of the options draft=false and draft=true

Table 7.1 *Label and description of the e- acceptor diimide cores studied.*

Id	Label	N	Formula	Diimide familly + 3D	
1	NDI	32	$C_{16}H_{10}O_4N_2$	} Naphthalene	
2	NDI_F4	38	$C_{16}H_6O_4N_2F_4$		
				X= H, F	
5	ADI	38	$C_{20}H_{12}O_4N_2$	} Anthracene	
6	ADI_F2	38	$C_{20}H_{10}O_4N_2F_2$		
7	ADI_CN2	40	$C_{22}H_{10}O_4N_4$		img/ADI.eps
8	ADI_Br2	38	$C_{20}H_{10}O_4N_2Br_2$		

7.3   Equations

Example 1

```
\begin{equation}
S_{\textrm{T}}(k) - 1 \equiv
\sum_{\alpha = 1}^n \sum_{\beta = 1}^n
\frac{c_\alpha c_\beta b_\alpha b_\beta}{\left< b \right>^2}
\left[S_{\alpha\beta}^{\textrm{FZ}}(k) - 1\right]
\end{equation}
```

$$S_{\text{T}}(k) - 1 \equiv \sum_{\alpha=1}^{n} \sum_{\beta=1}^{n} \frac{c_\alpha c_\beta b_\alpha b_\beta}{\langle b \rangle^2} \left[S_{\alpha\beta}^{\text{FZ}}(k) - 1 \right] \tag{7.1}$$

Example 2

```
\begin{equation}
E_\textrm{tot} =
\underbrace{\sum_{i=1}^{occ}  n_i \, \langle \psi_i |
- \frac{\nabla^2}{2} | \psi_i \rangle}
_T_\textrm{s}[\rho] \quad + \quad
\underbrace{
\frac{1}{2}
\int_\mathbb{R}^6
\frac{(\rho(\textbf{r}) + \rho^Z(\textbf{r}))
(\rho(\textbf{r}')+ \rho^Z(\textbf{r}'))}
{|\textbf{r}-\textbf{r}'|}
\, d\textbf{r} \, d\textbf{r}'}
_E_\textrm{POT}[\rho] \atop \quad + \quad
\underbrace{
\int_\mathbb{R}^3 \rho(\textbf{r})
\epsilon_\textrm{XC}[\rho] \, d\textbf{r}}
_E_\textrm{XC}[\rho]
\end{equation}
```

$$E_{\text{tot}} = \underbrace{\sum_{i=1}^{occ} n_i \langle \psi_i | - \frac{\nabla^2}{2} | \psi_i \rangle}_{T_{\text{s}}[\rho]} \quad + \quad \underbrace{\frac{1}{2} \int_{\mathbb{R}^6} \frac{(\rho(\mathbf{r}) + \rho^Z(\mathbf{r}))(\rho(\mathbf{r}') + \rho^Z(\mathbf{r}'))}{|\mathbf{r} - \mathbf{r}'|} \, d\mathbf{r} \, d\mathbf{r}'}_{E_{\text{POT}}[\rho]}$$

$$+ \quad \underbrace{\int_{\mathbb{R}^3} \rho(\mathbf{r}) \varepsilon_{\text{XC}}[\rho] \, d\mathbf{r}}_{E_{\text{XC}}[\rho]} \tag{7.2}$$

Partial glossary

8.1 Useful commands

8.1.1 In text mode

- **\aa**

  - The angström symbol: å

- **\AA**

  - The Angström symbol: Å

- **\footnote{**Here is my footnote**}**

  - To produce footnote(s):

  I want to insert**\footnote{**a footnote here**}**.
  I want to insert[1].

- **\par**

  - To start a new paragraph.

- **\caption**

  - To insert a caption: **\caption**[Small caption]**{**Full caption**}**

  The first optional argument is the name of the object for the list of figures or the list of tables, this is interesting if the caption of the figure or the table is long.

---

[1]a footnote here

8.1.2   In math mode

- **\bullet**

  - the symbol: •

- **\rangle**

  - the symbol: $\rangle$

- **\langle**

  - the symbol: $\langle$

- **\underbrace{**Equation**}_{**Information**}**

  To underbrace part(s) of an equation:

  \underbrace{y = a×x + b**}_\textrm{**Too simple !**}** $\underbrace{y = a \times x + b}_{\text{Too simple !}}$

  see equation 7.2 for more examples.

- **\lbrack**

  - the symbol: [

- **\rbrack**

  - the symbol:]

- **\left** + symbol in: {,(,[.. and **\right** + symbol in: },),] ...

  ensure that the symbols to be displayed immediately after the commands **\left** and **\right** will be large enough compared to the expression between **\left** and **\right**, see equation 7.1 for an example.

- **\sum_{**From**}^{**To**}**

  The sum symbol:

  \sum_{**i=0**}^{**N**} $\displaystyle\sum_{i=0}^{N}$

  see equations 7.1 and 7.2 for more examples.

- **\prod_**{From}^{To}

  The product symbol, the usage is similar to the **\sum** command:

  \prod_{i=0}^{N}
  $$\prod_{i=0}^{N}$$

- **\int_**{From}^{To}

  The integral symbol, the usage is similar to the **\sum** and the **\prod** commands:

  \int_{i=0}^{N}
  $$\int_{i=0}^{N}$$

  see equation 7.2 for more examples.

8.2 Useful environments

- center

  - definition: To insert centered text.
  - usage: **\begin{**center**} \end{**center**}

- flushright

  - definition: To insert right centered text.
  - usage: **\begin{**flushright**} \end{**flushright**}

- flushleft

  - definition: To insert left centered text.
  - usage: **\begin{**flushleft**} \end{**flushleft**}

- table

  - definition: To insert a floating table.
  - usage: **\begin{**table**}**[htbp!] **\end{**table**}

- tabular

  - definition: To use a column/row based layout.
  - usage: **\begin{**tabular**} \end{**tabular**}

- `equation`

  - definition: To insert an equation.
  - usage: **\begin{**equation**}**[htbp!] **\end{**equation**}**
  It is not possible to use the `equation` environment inside the `tabular` environment, use the $ $ or $**\displaystyle** $ commands instead.

- `figure`

  - definition: To insert a floating figure.
  - usage: **\begin{**figure**}**[htbp!] **\end{**figure**}**

- `itemize`

  - definition: To insert a list
  - usage: **\begin{**itemize**} \end{**itemize**}**

- `enumerate`

  - definition: To insert a numbered list.
  - usage: **\begin{**enumerate**} \end{**enumerate**}**

- `minipage`

  - definition: To insert a new content area, it is not mandatory but you can specify between [ ] how the text is centered in the minipage, `c` = centered, `l` = left, `r` = right, `t` = top, `b` = bottom. Then define the size of the frame, between { }.
  - usage: **\begin{**minipage**}**[clrtb]**{**?cm**} \end{**minipage**}**

- `bibliography`

  - definition: To insert a bibliography.
  - usage: **\begin{**bibliography**} \end{**bibliography**}**

- `landscape`

  - definition: To insert a page in landscape view, even if the document is in portrait view.
  - usage: **\begin{**landscape**} \end{**landscape**}**

- `picture`

  - definition: To draw using LaTeX commands ... no idea how it works !
  - usage: **\begin{**picture**} \end{**picture**}**

Error messages

The LaTeX compilation process can generate 4 types of messages:

- Information messages about the compilation

- Warning messages TeX and LaTeX

- LaTeX syntax error messages

- TeX syntax error messages

In the following I will only focus on few of the main the error messages.
When the compilation fails and you receive an error message, the compiler will stop and enter the interactive mode. Several actions are then possible pressing a letter of the keyboard (followed by `Return`):

`X`	to stop the compilation
`H`	to get more information about the error
`Return`	to ignore the error and continue (if possible)
`R`	to ignore all coming errors
`E`	to edit the current file
`?`	to list all possibilities

9.1 LaTeX errors

- `! Bad use of \\.`

  You used the **** command between 2 paragraphs, delete it or replace it by **\vspace**

- `! There's no line here to end.`

  You used the **** command between 2 paragraphs, delete it or replace it by **\vspace**

- ! **\begin{...}** ended by **\end{...}**.

  The environment opened by the **\begin** command was closed by the **\end** command of another environment.

- ! Can be used only in preamble.

  A command was found after the **\begin{**document**}** but it has to be located in the preamble.

- ! Cannot determine size of graphic in ... (no Bounding Box).

  The graphic file to be included do not specify its dimensions. Check the file name, or, try natwidth and natheight options for the **\includegraphics** command.

- ! Command name ... already defined.

  The command that you try to define using **\newcommand** already exists, you should use **\renewcommand** instead.

- ! Command ... invalid in math mode.

  The command cannot be used in math mode, use it outside or find the equivalent for the math mode.

- ! Environment ... undefined.

  The name of an environment passed to the command **\begin** is unknown. Correct the name of the environment or define it.

- ! File ... not found.

  The file can not be found by LaTeX, check the file name and that it is available.

- ! Missing **\begin{**document**}**.

  The command **\begin{**document**}** was not found in the source file, insert it at the appropriate location.

- ! Something's wrong-perhaps a missing **\item**.

  You probably forgot to use a **\item** command in a list environment (like itemize or enumerate). In other words the list is empty.

- ! Too many unprocessed floats.

  There are too many floating objects (figures, tables ...) on a single page.

- ! Undefined color '...'.

  You need to define the color you want to use using the **\definecolor**.

- ! Unknown graphics extension ....

  The graphic file to be included by the **\includegraphics** command is unknown. Check the file name or change the file format.

- ! Option clash for package ....

  A package was loaded twice with different options. Correct the preamble to load the package only once with the 2 options, or a single one if the options are conflicting.

9.2 TEX errors

- ! Extra alignment tab has been changed to **\cr**.

  There are too many column separators ,&, in a tabular environment. Delete the extra & or add column.

- ! Extra } or forgotten $.

  The command to enter and exit the math mode, $, do not match, there is an extra or a missing $.

- ! Missing number, treated as zero.

  A command that was supposed to receive parameter(s) did not receive any.

- ! Misplaced alignment tab character &.

  A character & was used outside a tabular environment. Delete it or replace it by **\&**.

- ```
  !  Missing { inserted.
  ```

  ```
  !  Missing } inserted.
  ```

  There is probably a { or a } missing, or a syntax error before the line of the error.

- ```
  !  Missing $ inserted.
  ```

  A math command was used outside the math mode. Put the command between $.

- ```
  !  Undefined control sequence.
  ```

  TEX found an undefined command, check its name or define it.

Some useful links

- The LATEX project web page: https://www.latex-project.org/

- The TEX Users Group, **TUG**: https://www.tug.org/ The TUG was founded in 1980 to provide an organization for people who are interested in typography and font design, and/or are users of the TeX typesetting system invented by Donald Knuth.

- The Comprehensive TEX Archive Network, **CTAN**: http://ctan.org/ The CTAN is the primary repository for TEX-related software on the Internet. CTAN has many thousands of items, to find any information regarding TEX or LATEX you can use:
    - The CTAN search page: http://ctan.org/search.html
    - The CTAN TEX catalogue: http://texcatalogue.ctan.org/

  All existing LATEX packages are listed and can be downloaded from the CTAN website.

- The TeX Live web page: https://www.tug.org/texlive/

- The MiKTeX web page: http://miktex.org/

- The Texmaker web page: http://www.xm1math.net/texmaker/index_fr.html

- The TEXnicCenter web page: http://www.texniccenter.org/

- The TeXworks web page: http://tug.org/texworks/

- TeXstudio web page: https://sourceforge.net/projects/texstudio/

- The Mendeley web page: https://www.mendeley.com/

- The LATEX Wikibooks: https://en.wikibooks.org/wiki/LaTeX

Bibliography

[1] Christian Rolland. *LATEX par la pratique*. O'Reilly, 1999. (On page 1.)

[2] S. Le Roux. *Basic tutorial to BASH programming*. 2016. (On pages 12 et 39.)

The text editor "gVim"

The GNU image manipulation program "GIMP"

And the document preparation system "$\math鱼{LaTeX}\,2_\varepsilon$".