LaTeX tutorial

Dr. Robin Roche

August 24, 2016

Contents

1 Getting started

LaTeX is a free document preparation system, that can be compared to Microsoft Word. One of the main differences is that it is not WYSIWYG, so the user only needs to focus on the content. Formatting is done automatically, and can be easily changed. It is commonly used for typesetting theses and publications in many scientific research fields.

This tutorial is a guide designed to help Ph.D. students get a basic understanding of how LaTeX works and can be used, with the goal of writing journal and conference papers. In the following, I assume that you have installed:

- A LaTeX distribution, such as TeXLive (use the full installation),
- An editor, such as Texmaker,
- A reference management software, such as JabRef.

Once these software are installed, you may want to download a paper template, e.g., from IEEE (journals or conferences) or Elsevier. The best approach is to open the template folder, open the appropriate `.tex` file with Texmaker, and fill-in the gaps with your paper content or replace dummy values with the appropriate ones (e.g., for authors and affiliations).

A `.tex` file is divided into three main parts, as shown in the outline of an IEEE paper below:

- The first part, until `\begin{document}`, is called the preamble. This is where the "parameters" of the paper are set: style, font size, margins, etc. This is also where packages, i.e., small pieces of software usually written by contributors to add new functionalities, are loaded. Here, we load the packages `cite` and `amsmath`.
- The second part, until `\maketitle`, is where you specific the title of the paper, list the authors and their affiliations, their emails, etc.

- The last part, until the end of the document, contains the actual content of the paper, i.e., what you are going to write, including the references and possibly the biographies.

```
\documentclass[journal]{IEEEtran}
\usepackage{cite}
\usepackage[cmex10]{amsmath}

\begin{document}

\title{Bare Demo of IEEEtran.cls for Journals}
\author{...}

\maketitle

\begin{abstract}
The abstract goes here.
\end{abstract}

\section{Introduction}

...

\bibliographystyle{IEEEtran}
\bibliography{bib_file}

\end{document}
```

In Texmaker, the simplest way to obtain an output file, i.e., a PDF file, is to compile the `.tex` file using `PDFLaTeX`. Another, more complicated, way is to compile using `LaTeX`, then `dvips` and finally `ps2pdf`. In addition to that, using `BibTex` may be required to include bibliography items correctly. Compiling several times is typically required so that all references are properly updated.

2 Text

The contents of a paper is organized in a hierarchical structure of a paper as follows:

1. Sections,
2. Subsections,
3. Subsubsections,
4. Paragraphs.

These different levels can be used as follows in the `.tex` document:

- To add a section, use `\section{Your section title}`.
- Similarly, for subsections, use `\subsection{Your subsection title}`.
- Each of these levels is numbered automatically.
- Do not use `\par` to separate paragraphs, just add an empty line between paragraphs.
- You can label sections to refer to them in the outline of the paper (see example below), at the end of the introduction with `Section~\ref{sec:model} describes...`.

  ```
  \section{System model}
  \label{sec:model}
  ```

You may create standard and numbered lists with the `itemize` and `enumerate` environments. The `\item` command enables starting a new item line. Two simple examples may be seen below.

```
\begin{itemize}
\item First line in the list.
\item Second line in the list.
\end{itemize}
```

- First line in the list.
- Second line in the list.

```
\begin{enumerate}
\item First line in the list.
\item Second line in the list.
\end{enumerate}
```

1. First line in the list.
2. Second line in the list.

A lot of other features exist. Here are some examples:

- To write in a color that is not black, the default color, use the packages `color` or `xcolor`, and the instruction `\textcolor{red}{my text}`.
- To write in bold, use `\textbf{your text}`.
- To write in italics, use `\textit{your text}`.
- You can add empty spaces and lines pretty much anywhere to facilitate reading. These lines or spaces are discarded when compiling so use them to avoid dense code. To add a non-breaking space or multiple spaces, use the sign `~`.
- You can add comments in code using `% Your comment`. These comments are not displayed in the output document.
- Depending on the available packages and on file encoding, adding accents may be difficult. To circumvent some issues, you can manually input letters with accents, such as in université (`universit\'e`) or À (`\`A`).
- To write the € symbol, use the `eurosym` package with `\EUR{}`.
- Quotation marks are input as in this "example" (`` ``example'' ``).
- Text exponents can be written as in 21st (`21\textsuperscript{st}`) century.

3 Equations

A strong point of LaTeX is that it enables typesetting complex equations and displaying them beautifully. Basic equations are implemented as follows, with the corresponding output below:

```
\begin{equation}
y = a\cdot x + b
\label{eq:y}
\end{equation}
```

$$y = a \cdot x + b \tag{1}$$

Like figures and tables, equations are automatically numbered. To remove the numbering from an equation, use `\begin{equation*}` and `\end{equation*}`. You can also label equations to refer to them, for example with `Equation~(\ref{eq:y}) shows that...`

The content of these equations can be typeset using a very wide variety of commands. Use the `amsmath` package to enable advanced features.

- In equations, parentheses, brackets, braces and similar signs are automatically sized if you use the right command, i.e., `\left(` and `\right)`, not `(` and `)`. Similar commands exist for brackets (`\left[`), braces (`\left\{`), etc.

```
\begin{equation}
(1 + \dfrac{n}{2}) \quad \text{vs.} \quad 1 + \left( 1 + \dfrac{n
    }{2} \right)
\end{equation}
```

$$(1 + \frac{n}{2}) \quad \text{vs.} \quad 1 + \left(1 + \frac{n}{2}\right) \tag{2}$$

- Numerous symbols exist such as:
  - $\forall$ (`\forall`),
  - $\in$ (`\in`),
  - $\exists$ (`\exists`),
  - $\leq$ / $<$ / $=$ / $\neq$ / $>$ / $\geq$ (`\leq / < / = / \neq / > / \geq`),
  - ω/Ω (`\omega / \Omega`),
  - $\pm$ (`\pm`),
  - $\approx$ (`\approx`).

- Similarly, operators and other functions must be input using the corresponding command. For example, for multiplications, use $\times$ (`\times`) or $\cdot$ (`\cdot`), not $*$ ($*$). Fractions are implemented with `\dfrac{a}{b}`, as in the example below.

```
\begin{equation}
\min_{S^*} \left\{ \Phi + \cos \left(\dfrac{\alpha}{n}\right) + \
    sinh \theta \right\}
\end{equation}
```

$$\min_{S^*} \left\{ \Phi + \cos \left(\frac{\alpha}{n}\right) + \sinh \theta \right\} \tag{3}$$

```
\begin{equation}
\lim_{t\rightarrow 0} f(t) + \int_{t=0}^{\infty} g(t) = \sum_{k
    =0}^{\infty} h(k)
\end{equation}
```

$$\lim_{t \rightarrow 0} f(t) + \int_{t=0}^{\infty} g(t) = \sum_{k=0}^{\infty} h(k) \tag{4}$$

- Powers and indices can easily be typeset using the `^` and `_` signs, as in the example below. Square roots are implemented with a dedicated function.

```
\begin{equation}
P_{1} + P_{2}^{3t} = \sqrt{3} + \sqrt[3]{15}
\end{equation}
```

$$P_{1} + P_{2}^{3t} = \sqrt{3} + \sqrt[3]{15} \tag{5}$$

- To add text in equations, use `\text{your text}`, otherwise text will not be displayed properly, as shown in the example below. Text can also be inserted as indices or powers, but with `P_{\mathrm{index}}` (output: P_{index}), for example, so the size of the text is automatically adapted.

```
\begin{equation}
1 + Text example \quad \text{vs.} \quad 1 + \text{text example}
\end{equation}
```

$$1 + Textexample \quad \text{vs.} \quad 1 + \text{text example} \tag{6}$$

- To include equations or variable names in text, use the `$` delimiters instead of the `equation` environment. For example, to mention variable α, we use `$\alpha$`.

- Spaces can be adjusted using, by order of increasing space, `\,`, `\:`, `\;`, `\quad`, and `\qquad`.

```
\begin{equation}
a \, b \: c \; d \quad e \qquad f
\end{equation}
```

$$a\,b\,c\,d\quad e\qquad f \tag{7}$$

- To add braces above or below some terms, use `\underbrace` or `\overbrace`, as in the example below.

```
\begin{equation}
\underbrace{P_{1} + P_{2} + P_{3}}_{= P} + Q = 100
\end{equation}
```

$$\underbrace{P_1 + P_2 + P_3}_{=P} + Q = 100 \tag{8}$$

- Dots can be added with the `\dots` command.

```
\begin{equation}
i=\{ 1, \dots, n \}
\end{equation}
```

$$i = \{1, \dots, n\} \tag{9}$$

Depending on what you want to implement, several environments may also be useful:

- To display an equation on several lines or a group of equations, use `align`. Note that this environment replaces the `equation` environment. Avoid using `eqnarray`, which is deprecated. All lines are aligned according to the location of the sign `&` in each line.

```
\begin{align}
f(x) &= (x+a)(x+b)\\
     &= x^2 + (a+b)x + ab
\end{align}
```

$$f(x) = (x+a)(x+b) \tag{10}$$
$$= x^2 + (a+b)x + ab \tag{11}$$

  The `align` environment is especially useful for long equations that need to be brocken down into several lines. An issue is however how to properly display (i.e., with the right size) parentheses, brackets,braces, etc. If you use a left opening, and the equation continues on another line, use `\right.` at the end of the first line, and `\left.` at the beginning of the second (see the example below). By default, all lines are numbered. To remove the numbering from a line, add `\nonumber` to this line.

```
\begin{align}
A &= \left( 1 + \dfrac{1}{2} + \dfrac{1}{3} + ... \right. \
   nonumber\\
  &\qquad + \left.\vphantom{dfrac{1}{2}} \dfrac{1}{7} + \dfrac
     {1}{8} + \cdots + \dfrac{1}{n} \dots \right)
\end{align}
```

$$A = \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \cdots + \frac{1}{n} \ldots\right) \tag{12}$$

- To display a matrix, use `bmatrix` or `pmatrix`. Matrix elements are separated with `&`, and lines with `\\`, as for tables.

```
\begin{equation}
A_{m,n} =
 \begin{bmatrix}
  a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
  a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
  \vdots  & \vdots  & \ddots & \vdots  \\
  a_{m,1} & a_{m,2} & \cdots & a_{m,n}
 \end{bmatrix}
\end{equation}
```

$$A_{m,n} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{13}$$

- To display an equation with multiple cases, use the `cases` environment.

```
\begin{equation}
f(n) =
\begin{cases}
n/2       & \quad \text{if } n \text{ is even}\\
-(n+1)/2  & \quad \text{if } n \text{ is odd}\\
\end{cases}
\end{equation}
```

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases} \tag{14}$$

The Wikibook on LaTeX contains a lot of examples with equations, especially the Mathematics and Advanced Mathematics pages. Also take a look at the documents listed in the references for more information, a longer list of symbols, etc.

4 Citations

LaTeX has a very convenient citation utility, called `BibTeX`. It can be used as follows with JabRef:

1. In Google Scholar or any other source, obtain the `BibTeX` code containing the information for the document you want to cite. It should be similar to the example below. Note that the automatic formatting may not be appropriate, and that you may need to correct it.

```
@article{hansen2015heuristic,
  title={Heuristic optimization for an aggregator-based resource
     allocation in the Smart Grid},
  author={Hansen, Timothy M and Roche, Robin and Suryanarayanan,
     Siddharth and Maciejewski, Anthony and Siegel, Howard Jay},
```

```
  journal={IEEE Transactions on Smart Grid},
  volume={6},
  number={4},
  pages={1785--1794},
  year={2015},
  publisher={IEEE}
}
```

2. In JabRef, create a new entry in your `.bib` file (saved in the same folder than the `.tex` file) and paste the code you just obtained for the document. Copy the identifier of the document. In the above example, the identifier is `hansen2015heuristic`.
3. To cite the document in your own paper, use `\cite{hansen2015heuristic}`. To cite multiple papers at the same time, use `\cite{hansen2015heuristic,roche2013multi}`. The list of references at the end of the paper is generated automatically.
4. For IEEE papers, you must point to the `.bib` file where you added the entries with JabRef, as in the example below where `bib_file` is the name of the `.bib` file. You may need to run `BibTeX` to generate and update the list of references.

```
\bibliographystyle{IEEEtran}
\bibliography{bib_file}
```

5 Figures, tables and algorithms

Figures, tables and algorithms are called floats, and are numbered automatically. All floats must have a label, and you must refer to all floats in the text.

Figures

To add a figure in the paper, copy the following code and customize it according to your figure. The output of this sample code is shown in Fig. 1.

```
\begin{figure}[!ht]
        \centering
        \includegraphics[width=0.15\columnwidth]{figures/algorithm_
            flowchart.pdf}
        \caption{Flowchart of the proposed algorithm.}
        \label{fig:algo_flowchart}
\end{figure}
```

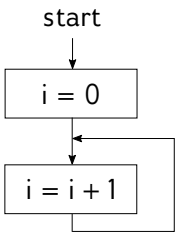

Figure 1: Flowchart of the proposed algorithm.

- You can define how you would *prefer* the figure to be inserted in the text. Here, we use `!ht` to say that we would prefer the figure to be inserted where it is in the code (`!h`), and otherwise at the top of a page (`t`). One may also use `b` for the bottom of a page, and `p` for a page only containing floats. Actually, LaTeX automatically adds the figure where it is the most appropriate; do not try to influence this other that through these instructions.

- You can define the figure size in multiple ways, by specifying a width, a height, or a scaling factor. Here, we specify a given width with `width=0.15\columnwidth`, which means that we want the figure to be 15% the width of the current column (in double-column papers, the coefficient is usually set to 1.0 to use the entire width of the column). But you can specify other sizes, relative to page width (`width=0.8\textwidth`), height (`height=5cm`), or to a scaling factor (`scale=0.7`).
- The file is given relatively to the folder where the `.tex` file is located. Use a name that describes the figure, not a number. For example, avoid `fig12.pdf`, as the figure number may change if you add or remove another figure!
- Add a caption to describe the content of the figure. This caption should be sufficient to understand what is shown in the figure.
- You can specify a label for the figure, with `\label{fig:algo_flowchart}`. You are free to choose the label you want, but I suggest starting it with `\label{fig:}` for figures. Choose a descriptive label, because you will need to use it when refering to the figure in the text with instructions like `As shown in Fig.~\ref{fig:algo_flowchart}`.
- If you want the figure to span all columns, i.e., have a figure that uses the width of the entire page, use `\begin{figure*}` and `\end{figure*}`.

As a general rule, I encourage you to use vector formats (`.pdf` or `.eps`), except for photographies (`.jpg`). Format `.png` may also be acceptable for diagram and curves, if figures are exported in high resolution.

To export a Matlab plot for use in LaTeX, do as follows:

1. Plot the graph, and add the required labels (labels on the x and y axes with units, title).
2. Save it in `.pdf` and `.fig` (so you can modify it later if necessary).
3. Open the `.pdf` file with Inkscape, to adjust the margins.
4. Go to: File > Document properties, and click on 'Resize to content'.
5. Click on 'Resize page to drawing or selection'. Do not change the margins manually.
6. Save the file.

Tables

To add a table in the paper, copy the following code and customize it according to your table. The output of this sample code is shown in Table 1.

```
\begin{table}[!ht]
\renewcommand\arraystretch{1.3}
\centering
\begin{tabular}{|l|c|c|}
\hline
Column 1 & Column 2 & Column 3 \\
\hline
A1 & \multicolumn{2}{|c|}{A2+A3} \\
\hline
B1 & B2 & B3 \\
\hline
\multirow{2}{*}{C1+D1} & C2 & C3 \\
\cline{2-3}
 & D2 & D3\\
\hline
\end{tabular}
\caption{Table example with cells spanning multiple columns and rows.}
\label{tab:table_example}
\end{table}
```

Column 1	Column 2	Column 3
A1	A2+A3	
B1	B2	B3
C1+D1	C2	C3
	D2	D3

Table 1: Table example with cells spanning multiple columns and rows.

- You can define how you would *prefer* the table to be inserted in the text. Options are the same than for figures.
- The columns are defined with the instruction `|l|c|c|`. Here, we want the first column to be aligned left (`l`), and the other two centered (`c`). The vertical lines (`|`) before and after the alignment identifiers are used to determine whether or not a vertical line should be drawn.
- The width of the columns is determined automatically.
- For each line in the table, use `&` to separate cells horizontally, and `\\` at the end of a line. The number of cells in a line must match the number of columns.
- To add a horizontal line, use `\hline`.
- To increase row height, add the following instruction before the table starts: `\renewcommand \arraystretch{1.3}`. You can tune the desired height by changing the scaling factor, here `1.3`.
- To have a cell spanning multiple columns, we use `\multicolumn{2}{|c|}{A2+A3}`, where we want the cell to span 2 columns, with the content (`A2+A3`) centered between vertical lines.
- However, to have a cell spanning multiple rows, you must use the package `multirow`, with `\usepackage{multirow}`. Here, `\multirow{2}{*}{C1+D1}` means that the cell should span 2 rows, the width is automatically determined (`*`, and the content of the cell is `C1+D1`. You can add a partial horizontal line with `\cline{2-3}`, i.e., a line spanning columns 2 to 3.
- As for figures, add a caption to describe the content of the table and a label to refer to it. You are free to choose the label you want, but I suggest starting it with `\label{tab:}` for tables. Note that in IEEE papers, the caption is above the table, not below.
- If you want the table to span all columns, i.e., have a table that uses the width of the entire page, use `\begin{table*}` and `\end{table*}`.

Algorithms

To include an algorithm pseudocode in a paper, packages `algorithmic` and `algorithm` may be used. Copy the following code example and customize it. The output of this sample code is shown in Algorithm 1.

```
\begin{algorithm}[!ht]
\caption{Pseudocode of the algorithm to compute $y = x^n$.}
\label{alg:x_power_n}
\begin{algorithmic}
    \REQUIRE $n \geq 0 \vee x \neq 0$
    \ENSURE $y = x^n$
    \STATE $y \Leftarrow 1$
    \IF{$n < 0$}
        \STATE $X \Leftarrow 1 / x$
        \STATE $N \Leftarrow -n$
    \ELSE
```

```
        \STATE $X \Leftarrow x$
        \STATE $N \Leftarrow n$
    \ENDIF
    \WHILE{$N \neq 0$}
        \IF{$N$ is even}
            \STATE $X \Leftarrow X \times X$
            \STATE $N \Leftarrow N / 2$
        \ELSE[$N$ is odd]
            \STATE $y \Leftarrow y \times X$
            \STATE $N \Leftarrow N - 1$
        \ENDIF
    \ENDWHILE
\end{algorithmic}
\end{algorithm}
```

Algorithm 1 Pseudocode of the algorithm to compute $y = x^n$.

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
  $y \Leftarrow 1$
  if $n < 0$ **then**
    $X \Leftarrow 1/x$
    $N \Leftarrow -n$
  else
    $X \Leftarrow x$
    $N \Leftarrow n$
  end if
  while $N \neq 0$ **do**
    if N is even **then**
      $X \Leftarrow X \times X$
      $N \Leftarrow N/2$
    else {N is odd}
      $y \Leftarrow y \times X$
      $N \Leftarrow N - 1$
    end if
  end while

- For this to work, the `algorithmic` and `algorithm` packages must be loaded. Add the following instructions before `\begin{document}`:

  ```
  \usepackage{algorithmic}
  \usepackage{algorithm}
  ```

- Pseudocode is a way of describing an algorithm with text rather than code. It does not depend on a specific programming language. As pseudocode is for humans, not machines, some instructions are usually removed, such as variable declarations.
- Package `algorithmic` enables typesetting the pseudocode with instructions, such as statements, for/while loops, if/then/else, comments, etc. The list of the main instructions is as follows:

  ```
  \STATE <text>
  \IF{<condition>} \STATE {<text>} \ELSE \STATE{<text>} \ENDIF
  \FOR{<condition>} \STATE {<text>} \ENDFOR
  \WHILE{<condition>} \STATE{<text>} \ENDWHILE
  \REQUIRE <text>
  \ENSURE <text>
  \RETURN <text>
  \COMMENT{<text>}
  \AND, \OR, \XOR, \NOT, \TO, \TRUE, \FALSE
  ```

- Package `algorithm` creates an algorithm environment, so that such algorithms can be considered as another type of floats, like figures and tables. With this, algorithms are thus automatically numbered, with a caption and a label to refer to it. The options for specifying the prefered location of the float are also available here, as for figures and tables.

6 For further reading

Although these basic LaTeX instructions should be sufficient in most cases, almost everything can be modified manually but it may not be easy. A Google search is usually the best way to find a solution. The following websites are good resources:

- latex-community.org
- tex.stackexchange.com
- latex.wikia.com

You might also want to refer to more detailed tutorials for more information. The following documents and websites contain useful information related to LaTeX.

[1] Frank Mittelbach, Rainer Schöpf, et al.,
LaTeX – A Document Preparation System,
`https://www.latex-project.org/`

[2] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl,
The Not So Short Introduction to LaTeX2$_\epsilon$,
`https://tobi.oetiker.ch/lshort/lshort.pdf`

[3] Andrew Roberts,
Getting to Grips with LaTeX,
`http://www.andy-roberts.net/writing/latex`

[4] David R. Wilkins,
Getting Started with LaTeX,
`http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/`

[5] E. Krishnan et al. (Indian TeXUsers Group),
LaTeXTutorials: A Primer,
`https://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf`

[6] Claudio Vellage (latex-tutorial.com),
LaTeXTutorial – Interactive Lessons, Code Examples – For Beginners,
`https://www.latex-tutorial.com/`