LaTeX tips

Charles University SIAM Student Chapter

February 25, 2022

The following is a compilation of Nick Higham's **Top Tips for New LaTeX Users**, Mark Trettin's **An essential guide to LaTeX 2ε usage**, and our own tips & suggestions. Complete beginners should first see basics on **LaTeX Wikibooks**.

If you do not know how type something in LaTeX, generally you should follow these steps

1. Consult **LaTeX Wikibooks** or **Overleaf documentation**. Is there a standard solution to your problem?
2. Check one of the recommended references [1] or [2].
3. Google the solution (preferably use **LaTeX Stack Exchange**) and try to understand the solution using the sources mentioned above.

Note that this text applies to English. Most tips are universal, but some (fer example the rules for hyphens and dashes) are language-based and not completely true for Czech. A reliable source for Czech orthography, grammar and typography could be found at the **Internet Language Reference Book** of the Institute of the Czech Language.

New paragraphs, new pages

In LaTeX a new paragraph is started by leaving a blank line or equivalently by `\par` command. Do not start a new paragraph by using `\\` (it merely terminates a line). Indeed, you should almost never type `\\`, except within environments such as `equation`, `tabular`, etc.

Hyphen, en dash, em dash

There are three important horizontal dash-like characters

- **hyphen** - typed as `-`

  Typically used between the elements of compound words, such as "hard-boiled egg", typed as "`hard-boiled egg`", or in names, such as "Antoine de Saint-Exupéry", typed as "`Antoine de Saint-Exup\'{e}ry`".

- **en dash** – typed as `--`

  Used to join names corresponding to different people in a phrase, such as "Navier–Stokes equations", typed as "`Navier--Stokes equations`", or in ranges, such as "Sections 3–7", typed as "`Sections 3--7`".

- **em dash** — typed as `---`

  Used to "set aside a phrase—like this—in a sentence", typed as "`set aside a phrase---like this---in a sentence`".

Inline math mode

Always type mathematics in math mode as `$...$` or `\(...\)` to produce "$y = f(x)$" instead of "y = f(x)", and "the dimension n" instead of "the dimension n". Punctuation should appear *outside* math mode, for inline math expressions, otherwise the spacing will be incorrect

- **correct:** `The variables $x$, $y$, and $z$ satisfy $x^2 + y^2 = z^2$.`
- **incorrect:** `The variables $x,$ $y,$ and $z$ satisfy $x^2 + y^2 = z^2.$`

Displayed math mode

Do not use `$$...$$` for separating math formulae from text as this command modifies vertical spacing within formulae and is difficult to parse. Instead, use `\[...\]` or better yet, opt for `\begin{equation*}...\end{equation*}` in case you are using the `amsmath` package. If needed, the displayed equation can then be converted into a numbered one simply by omitting the star from `equation*`.

For displayed equations, punctuation should appear as part of the display. All equations *must* be punctuated, as they are part of a sentence.

Text in displayed math mode

When a displayed equation contains text such as "subject to $x \geq 0$", instead of putting the text in `\mathrm` put the text in a `\text` command (requires `amsmath` package). Note that `\text` switches out of math mode, and this has the advantage of ensuring the correct spacing between words. For example

```
\begin{equation*}
  f(x) =
  \begin{cases}
    0 & \text{if $x = 0$,}
    \\
    \frac{1}{x} & \text{otherwise.}
  \end{cases}
\end{equation*}
```

is rendered as

$$f(x) = \begin{cases} 0 & \text{if } x = 0, \\ \frac{1}{x} & \text{otherwise.} \end{cases}$$

Math expressions

Ellipses (dots) are never explicitly typed as "...". Instead, they are typed as `\ldots` for baseline dots, as in `$x_1, x_2, \dots, x_n$` (giving $x_1, x_2, \dots, x_n$) or as `\cdots` for vertically centered dots, as in `$x_1 + x_2 + \cdots + x_n$` (giving $x_1 + x_2 + \cdots + x_n$). When using `amsmath` package the command `\dots` tries to automatically make the choice for you and switches between `\ldots` and `\cdots` depending on the following symbol.

If you are using angle brackets to denote an inner product use `\langle x, y \rangle` which renders as $\langle x, y \rangle$ instead of `< x, y >` which results in $< x, y >$.

For "much less than", type `\ll`, giving $\ll$, not `<<`, which gives $<<$. Similarly, "much greater than" is typed as `\gg`, giving $\gg$.

Avoid the `\over` command for typing fractions as it is difficult to parse and clashes with the `amsmath` package. Instead, use `\frac{}{}` in displayed math mode.

Defining macros

Always use `\newcommand{\<name>}{...}` for defining new macros. Never use `\def\<name>{...}`. The main problem with `\def` is that no check is done on whether there already exists another macro of the same name. So a macro defined earlier may be overwritten without any error warning. Macros may be re-defined using `\renewcommand{\<name>}{...}`.

Functions and operators

Mathematical functions should be typeset in roman font. This is done automatically for the many standard mathematical functions that LaTeX supports, such as `\sin`, `\exp`, `\max`, etc. If the function you need is not built into LaTeX, such as a `\diag` operator, create your own by the general `\newcommand{\diag}{\operatorname{diag}}` or, when using the `amsmath` package, by `\DeclareMathOperator{\diag}{diag}`. Do not use `\newcommand{\diag}{\mathrm{diag}}` for defining operators as it renders the function name incorrectly.

Parentheses, brackets, braces, ...

When enclosing a math formula inside some delimiters (parentheses, brackets, braces, etc.) a good start is to always use the `\left` and `\right` commands with the appropriate delimiter. These commands change the delimiters' size dynamically depending on the content. For example

```
\begin{equation*}
  \sum_{k=0}^n a q^k = a \left( \frac{1 - q^{n + 1}}{1 - q} \right).
\end{equation*}
```

yields

$$\sum_{k=0}^{n} aq^k = a\left(\frac{1-q^{n+1}}{1-q}\right).$$

When the automatic scaling is too drastic, advanced users might want to tweak some cases manually using the commands, such as `\big`, `\Big`, etc. instead of `\left` and `\right`.

Centering and flushing floats

A frequently seen mistake is to use `\begin{center}...\end{center}` inside environments such as figure, table, minipage, etc. However, `center` can cause additional vertical space around the float. To properly center floats, use `\centering` command, such as in this example

```
\begin{figure}[ht]
  \centering
  \includegraphics{imagefile}
  \caption{Some caption.}
  \label{fig:figure}
\end{figure}
```

The command `\clearpage` tells LaTeX to print all the figures (collected on the way) here, and not to place them in the following text. This is useful e.g. when starting new section.

Overfull hbox

The `overfull hbox` warning means that a line of your document is too long to fit within the horizontal space on the page, and LaTeX couldn't find a good way to break it apart. If it is difficult to spot, `\documentclass[draft]{...}` will help you to locate the exact place where the text is hanging out past the margin. Adding the package `\usepackage{microtype}` should be sufficient to fix this problem.

Tables

If you find uncomfortable writing tables in LaTeXand if you are used to write data in Excel, we have great option for you! There is a plug-in called `Excel2LaTeX`, which can download **here**. You can just open your sheet in Excel as well as the file Excel2LaTeX.xla (which you have downloaded) and enable macros. Then, you should see a button `Convert table to LaTeX` under the tab Add-ins.

And even if you don't like Excel, you can create your table interactively, for example **here**, and simply generate the corresponding LaTeX code.

Labeling and referencing

In LaTeX sections, formulae, floats, etc. are marked using the `\label` command and referenced using the `\ref` command. However, when using the `amsmath` package one usually references *equations* using the `\eqref` command which adds parentheses around the reference number.

It is common practice among LaTeX users to add a few letters to the label to describe *what* you are referencing. For example use `\label{sec:some-section}` to denote sections, `\label{fig:some-figure}` to denote figures, `\label{eq:some-equation}` to denote equations, etc.

Note that in floats (figures, tables, etc.) the `\label` command comes *after* the `\caption` command otherwise you will get a reference to the (sub)section where float is declared instead of a reference to the float itself.

Splitting and aligning equations

When your equation is too long to be fitted on a single line, use `multline` environment from the `amsmath` package instead of `equation` and split the equation at a suitable place using `\\`. For example

```
\begin{multline}
  \label{eq:multline-equation}
  \sum_{i=1}^{30} x_i
  =
  x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15}
  \\
  + x_{16} + x_{17} + x_{18} + x_{19} + x_{20} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} +
        x_{28} + x_{29} + x_{30}.
\end{multline}
```

yields

$$\sum_{i=1}^{30} x_i = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{30}. \quad (1)$$

To align several numbered equations vertically at points marked with `&` use `align` environment from the `amsmath` package. For example

```
\begin{align}
  \label{eq:align-1}
  a & = 3 r^2 + \sin \varphi,
  \\
  \label{eq:align-2}
  b & = 2 \ln r - \cos \varphi.
\end{align}
```

is rendered as

$$a = 3r^2 + \sin\varphi, \quad (2)$$
$$b = 2\ln r - \cos\varphi. \quad (3)$$

For aligning several expression while numbering the whole formula as a unit, use `split` environment from the `amsmath` package inside the `equation` environment. For example

```
\begin{equation}
  \label{eq:split}
  \begin{split}
    a & = \left( r + 1 \right)^2
    \\
      & = r^2 + 2 r + 1.
  \end{split}
\end{equation}
```

gives

$$\begin{aligned} a &= \left(r + 1\right)^2 \\ &= r^2 + 2r + 1. \end{aligned} \quad (4)$$

Avoid using the standard LaTeX environment `eqnarray` as it does not produce as good results as the `amsmath` environments, nor is it as versatile.

Note that the environments `multline` and `align` also have their unnumbered equivalents `multline*` and `align*`.

The environment `align*` allows for multiple columns of equations using multiple & symbols. A good explanation of the environment's behaviour is not easy to find, so we include it here. Two equivalent points of view are possible

1. The odd &s in line represent the points of alignment, the even &s split the content.
2. All &s separate line into columns. Odd columns are right-aligned, even columns are left-aligned.

For example

```
\begin{align*}
  \text{ right }| & \text{ left } & \text{ right }| & \text{ left } & \text{ right }| & \text{ left } \ldots
  \\
  | & & | & & | & \ldots
\end{align*}
```

will produce

$$\begin{aligned} \text{right} \mid \text{left} \\ \mid \end{aligned} \qquad \begin{aligned} \text{right} \mid \text{left} \\ \mid \end{aligned} \qquad \begin{aligned} \text{right} \mid \text{left} \ \ldots \\ \mid \ldots \end{aligned}$$

This section provides just a basic overview of equation splitting and aligning. See **LaTeX Wikibooks** for a nice list of the various kinds of equation and align environments.

Subequations environment

The `subequations` environment from the `amsmath` package provides a convenient way to number equations in a group with a subordinate numbering scheme. For example

```
\begin{subequations}
  \label{eq:subequations}
  \begin{align}
    \label{eq:subequation-1}
    a &= 3 r^2 + \sin \varphi,
    \\
    \label{eq:subequation-2}
    b &= 2 \ln r - \cos \varphi.
  \end{align}
\end{subequations}
```

is rendered as

$$a = 3r^2 + \sin\varphi, \tag{5a}$$
$$b = 2\ln r - \cos\varphi. \tag{5b}$$

Note that the labels read (5a) and (5b) instead of (5) and (6).

Making comments in the pdf file

When collaborating with several people, or just working on a bigger project on your own, it may be useful to make some to-do notes that are easily recognizable in the compiled pdf file. The easiest thing to do is to color, underline or cross-out desired words with the help of the `color` or `ulem` packages. However, for a more systematic approach of writing your to-do notes, opt for the `todonotes` package and its `\todo` command.

For example

```
``One can easily prove \todo{How?} that the set of all prime numbers is infinite.
The set of prime numbers is an example of a \emph{countable} infinite set.
On the other hand, the set of real numbers is an example of an \emph{uncountable} infinite set.''
```

renders as

"One can easily prove that the set of all prime numbers is infinite. The set of prime numbers is an example of a *countable* infinite set. On the other hand, the set of real numbers is an example of an *uncountable* infinite set."

How?

The main advantage of the `todonotes` package is that the notes are more visible and easily removable by using the option `\documentclass[final]{...}`. Also, at any point in the document a list of all the inserted to-do items can be listed with the `\listoftodos` command.

For more details on the usage of `todonotes`, see the documentation of the package at CTAN.

References

[1] Donald E. Knuth. *The TeXbook.* Addison-Wesley Professional, 1986.

[2] Leslie Lamport. *Latex: A Document Preparation System.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.