

# Everything you always wanted to know about L<sup>A</sup>T<sub>E</sub>Xing an article and I grew tired of telling

version 3.16227766017

Patrick van der Smagt

January 21, 2023

You don't know who Donald Knuth is? Stop reading here. This document is not for you. But if you do, you may have more satisfaction from reading the T<sub>E</sub>Xbook than these few lines. Do so, please. Here may be a surprise for you: it is probably on your computer.

Why? It's simple. When you want people to read your article, you need to transport its contents. To do so, you want your audience to focus on the contents, and not on how the contents it packed. You need to make the form inconspicuous; it should not distract the reader. The more streamlined it is, the better the reader can concentrate. That's why we start a sentence with a capital letter.

## 1 Language

Now, English is a very weird European language. There are many style guides, traditionally set by the publishing houses (but less so these days). Many of them have small idiosyncrasies, and you're free to adhere to any of those (or even your own, if you feel bold enough). Key, however, is consistency.

Beware your spelling bugs. You can squeeze them before giving your document to someone else, it saves a few iterations and many irritations. Make sure you decide whether you use US English or British English (BE) spelling and try to be consistent. Some notable differences include the following words:

British English	US English
colour, flavour, humour, etc.	color, flavor, humor, etc.
centre, fibre, etc.	center, fiber, etc.
modelling, levelling, etc.	modeling, leveling, etc.
aeroplane, aluminium, etc.	airplane, aluminum, etc.

but this list is far from complete. See, e.g., a corresponding [wiki page](#).

In BE, you can also choose for the “-ise” rather than “-ize” ending, depending on which dictionary you adhere to. To my best of knowledge, Oxford uses “minimize,” “minimization,” etc. while Chambers dictates “minimise,” “minimising,” etc. Statistically, “-ise” is preferred in BE.

An interesting anomaly is that, in US English, punctuation is within quotes. I used that style in this document, but BE does not prefer that anymore.

## 1.1 ll

“All, till, full, and well,  
always have double l  
except in compounds.”

(So why does  $\text{\TeX}$  write `overfull`? And what about `underfull`? `Overfulfil`, or is it `overfulfill`? `Woeful`, all of this.)

## 1.2 a or an?

Everyone knows that “an” is used before a vowel, “a” before a consonant. But that refers to the *sound* that follows, not to the spelling! So please write, “modern physics discovered that the constant indicated by **an**  $h$  is no longer relevant for describing uncertainty . . .”

## 1.3 Dash

The dash rule in English is really very, very simple (I am talking hyphens here). Generally, adverbs and adjectives are connected by a dash. To help you refresh your grammar: an adjective describes a noun. An adverb describes an adjective.

So, for instance, when we write “the red-beaten man,” we refer to a man who was beaten until red. The adverb “red” describes a property of “beaten,” not of man. If, however, we write “the red beaten man” (or, more clearly, “the red, beaten man”), the poor chap was red before being beaten.

So, write “the two-dimensional manifold” since “the two dimensional manifold” would mean something else (though I would not know what a “dimensional manifold” is). And ignore my previous comment on “light-weight” vs. “lightweight”, as I was blatantly wrong there.

## 1.4 Caveat lector

Now, finally, you’d be silly to listen to the above. First of all, I am not a native speaker, at least not of the English language. I am not even close to an expert on English, or any language for that matter. I like to be conscious about its use, but am not even very good at it.

Second, the above rules may be rules, but English is not that strict. Being consistent throughout a piece of text is certainly advisable, since it reduces reader confusion. But take the above dashes-piece. Yes, it may be correct. But if there is no confusion, as in “two dimensional manifold,” you need not be pedantic about putting the dash there. You can, but leaving it out is not necessarily wrong.

If you want to read more on the matter, one of my favourite books is *Making a point* by David Crystal. By the way, his other books are not less excellent.

# 2 $\text{\LaTeX}$ typesetting

## 2.1 Commands

A quick one about commands. Many  $\text{\LaTeX}$  and  $\text{\TeX}$  commands take one or more arguments. How does  $\text{\TeX}$  know when a command ends? The rules, which can be redefined, are simple in normal (“plain”) mode: a command starts with a `\` followed by a single non-alpha character, or one or more alpha characters. So, `\!` is defined as being one character (after the backslash) long only, but `\ashdhf` only ends when the first non-alpha character is seen.

If an alpha command is followed by one or more spaces, these spaces are interpreted as an end-of-command character, and does not render a space in the text. So, when I write `\TeX is great` I don't get what I want.

You can use this rule to your advantage. For instance, `\frac2n` yields exactly the same as `\frac{2}{n}`, but `\fracn2` goes bitterly wrong. In math mode, why not write `$n^2$` rather than `$n^{2}$` (yes, `^` is also a special control character, as are `$\%&\{\}_\sim`. Which did I miss?)

## 2.2 Space control

It is worth understanding the spacing algorithm in  $\TeX$ . The intricacies go beyond this short section, but the principle is as follows.  $\TeX$  defines glue between words, sentences, etc. This glue can be compared with a spring, having a base length, which can be compressed by a certain amount, and which can be extended by a certain amount. Normally, two words are separated by some horizontal glue, as are two lines, by vertical glue.  $\TeX$  then goes on defining the length of a sentence, for instance, and of a page, which also have some flexibility (in the order of picas).

Since the amount of glue is not infinite, there are cases where  $\TeX$  cannot fill a page or line with the amount of glue available. You may find an example in Sec. 5.1. So,  $\TeX$  gives up and outputs an error message (`overfull \hbox`).

You really need to scan your log file for such errors. And repair them! There are a few things you can do. The best thing is to reformulate your sentence, or help  $\TeX$  break words with the `\-` command. If you are lazy, however, you can use `\tolerance10000` in the preamble of the document. For  $\TeX$ , 10,000 is the same as infinity, and this means that the corresponding glues can be extended up “infinity.” You will never end up with overfull hboxes – but you will probably get `underfull` ones!

Then: after a full stop *not* following a capital letter,  $\TeX$  assumes the end of a sentence and adds more glue (i.e., more stretch). Sometimes you don't want this, and you can prevent this extra glue with a `\` (followed by a space) or a `~` (*not followed by a space*). The latter indicates a non-breaking space. To make things more intuitive, this also holds beyond a `”` or a `)`. So, type `I think, viz.\ I am` to yield “I think, viz. I am” rather than “I think, viz. I am.” Do you notice the difference? I do. Depending on how much the space has to be stretched to fill a line,  $\TeX$  will do that after a full stop.

`\raggedright` and `\raggedbottom` switch off  $\TeX$ 's line and page filling algorithms. The latter is, by default, off in article mode.

**To paragraphs.** Creating a new paragraph in  $\TeX$  is simply done by leaving a line blank. The previous line will be terminated, a new paragraph will be started after a length of `\parskip` and indented by `\parindent`. Use this. If you really cannot use the indent on the next line, don't use `\\` but instead write `\noindent` before the next paragraph. Use `\\` only if you *really* want a newline.

Of course, don't create a new paragraph when, for instance, your text goes on with a displayed equation! It will add spurious space, your fault.

## 2.3 Dashes

Watch the use of dashes. For numerical ranges, use the en-dash `--`.

As to em-dashes? Well, in 2020/2021 I changed my mind. I always adhered to the more US-centric approach of using an em-dash as an alternative to a comma—not pretty sometimes, but helps, and in  $\TeX$  is typed as `---`. Always without spaces around it!

But that is less customary in British English – and in quite a few other languages I know in fact – and for British, the Oxford style guide says, do not use the em-dash. Instead – as in this example – use the en-dash separated by spaces. In  $\text{\TeX}$ , it's typed with two hyphens.

In equations, the dash is to be transformed to a minus. So never write  $-1$  yielding  $-1$ , instead use  $\$-1\$$  giving  $-1$ .

In English, no spaces around dashes please, except of course preceding a minus sign.

## 2.4 Dots

Use  $\backslash\text{dots}$  for  $1, 2, \dots, n$ , I'm sure<sup>1</sup>.  $\backslash\text{dots}$  is smart: it adapts depending on the context. You can therefore also use it for  $1 + 2 + \dots + n$ , written as  $[1+2+\backslash\text{dots}+n]$ . You can always revert to  $\backslash\text{ldots}$  or  $\backslash\text{cdots}$  to take over control. You can even use  $\dot{\phantom{x}}$  ( $\backslash\text{vdots}$ ). In math mode there is also  $\ddot{\phantom{x}}$  ( $\backslash\text{ddots}$ ). I'm not aware of any other dots.

## 2.5 Quotes

British English prefers single quotes; US English double. But you can do double in BE documents (as I did in this document). Whatever you do – be consistent throughout your text!

To type quotes in  $\text{\LaTeX}$ , use  $\text{'}$  and  $\text{'}$  (or, of course, their single variants). The more general  $\backslash\text{ql}$  (for left quote) and  $\backslash\text{rq}$  will render the right quotes in the Babel language that you are typesetting. Alternatively, you can use  $\text{"}$  and  $\text{"}$ —but only if the Babel package is loaded.

If you did use Babel and set your language right, there is a nice automation for quotes in  $\text{\LaTeX}$  – as for many things, of course. Use

```
 $\backslash\text{usepackage}[\text{british}]{\text{babel}}$ 
...
 $\backslash\text{usepackage}[\text{autostyle}]{\text{csquotes}}$ 
 $\backslash\text{MakeOuterQuote}\{\text{"}$ 
```

or, if you did not specify the language of your document,

```
 $\text{usepackage}[\text{autostyle}=\text{false}, \text{style}=\text{british}]{\text{csquotes}}$ 
 $\backslash\text{MakeOuterQuote}\{\text{"}$ 
```

after which you can write  $\backslash\text{textsl}\{\text{She told me, "I bloody like this option".}\}$  to render *She told me, 'I bloody like this option'*.

## 3 Figures and tables

Most authors cannot drive  $\text{\LaTeX}$  well in placing these. In fact,  $\text{\LaTeX}$  offers a very nice set of macros for placing and naming these. You need just write  $\backslash\text{begin}\{\text{figure}\} \dots \backslash\text{end}\{\text{figure}\}$  and remember that the label must always come *after* the caption, and you are fine. Same for tables, but here the caption should *precede* the table contents.

Now, since the table caption comes before the table and the figure caption after the figure, typesetting rules dictate that, where possible, figures are at the top of a page and tables at the bottom.  $\text{\TeX}$  tries to do that for you, and in most cases you need never to worry about this anymore.

---

<sup>1</sup>a bit tricky: the sentence ends with  $\backslash\text{dots}\backslash$  to make sure there is a bit of space after the dots.

Please refrain from using these `[hbt p!]` position modifiers if you don't need to. By default,  $\text{\TeX}$  does a very good job in placing them. Also, please refrain from putting a figure in the middle of a column, unless you have a very good reason!

Rules aiming at reducing reader distraction ask you to refer to figures with `Fig.\ref{...}` and to tables with `Tab.\ref{...}`.

### 3.1 Table formatting

Even I am daunted by the range of possibilities to format a table in  $\text{\LaTeX}$ . The standard set of macros is not very good, but the number of extensions that are available – even if one only counts the good and useful ones – is challenging. See the below as my personal taste. That said, one may still think that typesetting tables is about  $\text{\LaTeX}$ 's worst feature. I know of no alternatives.

The issue of table formatting is, to a large extent, related to taste and design rather than typography – yet the line is thin. A succinct text on the matter can be found behind this link.

Apart from the looks, there are a few packages that you may need when working with more-than-trivial tables in  $\text{\LaTeX}$ . These include:

```
\usepackage{longtable} % must be loaded to use {longtabu}
\usepackage{tabu} % more general tabular environment, use longtabu for long tables
\usepackage{numprint} % formats numbers in a nice way
\usepackage{spreadtab} % allows calculations in tables, uses numprint for printing output
\usepackage[table]{xcolor} % alternative to tabu to create coloured cells
\usepackage{booktabs} % to make prettier tables, use \toprule...\midrule...\bottomrule
```

You only need `spreadtab` if you want to compute values in cells. `Tabu` gives a good alternative to the myriad of `tabularx`, `tabulary`, `longtabular`, and all of these older versions, plus does many things one does not need. `Numprint` prints scientific notation, is a good package to pretty-print tables with numbers, and to format the output of `spreadtab`. `xcolor` may be needed to colour cells, but `tabu` does it, too; so if you use `tabu`, don't use `xcolor`.

Check the manuals of these packages to enjoy their strength. A separate file `table-example` in this directory gives an example complex table.

## 4 Citing

There are two things to correct citing in  $\text{\LaTeX}$ . First, there is putting the cite in the paper. Quite an easy thing to do, by typing `\cite{ref1,ref2,ref3}`. Note that multiple cites are all included in one cite command.

The  $\text{\BibTeX}$  entry itself must be carefully constructed. Many people like to use bib management programs, such as the public-domain tool `JabRef`. Fine, but you still must feed it with the right info:

- for a journal publication, you must add title, author, year, volume number, pages, and possibly issue number. Pages are written as 5--10, but if you write 5-10 in a  $\text{\BibTeX}$  entry, it will automatically translate that to 5--10 for you. Under no circumstance use three dashes!
- for a conference publication, don't add info on where the conference was. The `address` field is to specify the address of the publisher house, and quite irrelevant for this kind of publication.

If you have them, specify page numbers. Then, of course, title, authors, name of the conference.

Now that may all be nice and well. If you want to be a bit more serious and flexible about citations – something you will probably need to do in a book or other long document with chapters – you will want to look at the quite exquisite Bib $\text{\LaTeX}$  package. It gives you a whole new experience of making bibliographies in  $\text{\LaTeX}$ . It is too mighty to deal with here.

## 5 Equations

### 5.1 Referring to equations

When having a displayed equation such as

$$i\hbar\frac{\partial\psi}{\partial t} = \frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi \quad (1)$$

we can say that thanks to Schrödinger we know Eq. (1) which is typed as `Eq.~(\ref{eq:schroedinger})`. The above line creates an overfull warning. How can we deal with that, and how should we?

Personally I got used to doing things by hand, but many people like packages to assist them in such (and other) references. `cleveref` is one of the more complete ones and worth checking out.

### 5.2 Punctuation in equations

These things are easy to answer, as easy as

$$\pi r^2.$$

If an equation, for instance

$$E = mc^2,$$

is part of a sentence, of course normal punctuation must be applied. It makes no sense adding a comma after an equation such as

$$f(\vec{x}) = \sum_i x_i$$

when a comma has no place there.

Oh, of course, punctuation should be within a display equation but outside an inline equation.

But there are some caveats. For instance, the comma in equations is a punctuation symbol. To change that, put it in braces to say, for instance, `$10\{, \}000\,|x|$ is not less than $|x|$`. Done without the braces:

$$10,000|x| \text{ is not less than } |x|$$

which is not as nice as the version with accolades:

$$10,000|x| \text{ is not less than } |x|$$

Note that, in both cases, I put a bit of extra space after the last 0, or else it looks too cramped. Accolades, braces, curly brackets... all the same thing, in this context.

### 5.3 Give them space

Quite a tricky business, getting spacing in equations right. But doing it right will tremendously improve the readability of equations, which is key to have your reader follow your writing.

In general, put in a bit of space before a derivative symbol: `$a\,dx/dy$` is right. Also, deleting the space in `$x^2!/2$` gives  $x^2/2$ , much nicer than  $x^2/2$ . As is adding some in `$\sqrt{2}\,x$` to yield  $\sqrt{2}x$  (compare  $\sqrt{2}x$ ).

### 5.4 Units

Units are, in general, written according to ISO style and that means, you use “s” for seconds, “m” for metres, etc. Make them roman (meaning, not slanted, not italics) and separate them from the number by half a space. So you write `I am 1.86\,m long, about 1,451,673,446,400\,ms old, and vibrate at 1\,Hz` to yield “I am 1.86 m long, about 1,451,673,446,400 ms old, and vibrate at 1 Hz” and don’t use math mode!

But, we write  $75\%^2$  And  $25^\circ\text{C}$  as `$25^\circ\text{C}^3$`.

There’s a nice package `siunitx` for assisting in these things.

### 5.5 italics or roman?

By default, equations are built from single-letter symbols, e.g.,  $f(x) = ax + b$ . But sometimes it makes sense to write variables in a more wordy fashion. People sometimes like to write  $vel = dpos/dt$ . As you can see from the typesetting, this goes horribly wrong! If you really need to write multi-letter variables describing some physical quantity, use `\mathrm`, as in `$\mathrm{vel} = d\,\mathrm{pos} / dt$` to render  $vel = d\,pos/dt$ .

### 5.6 Multiplying

One sees many symbols for scalars multiplication. There is the `\times` displayed as  $\times$ , or the `\cdot` aka  $\cdot$ , sometimes even `\ast`, i.e.,  $*$ .

Just don’t. For engineers:  $\cdot$  is, in general, the inner (or dot) product,  $\times$  the outer product, and  $*$  means convolution<sup>4</sup>. To write  $a$  times  $x$ , just type `$ax$` yielding  $ax$ .

### 5.7 Fractioning

Sure, you use `$\frac{1}{2}$` to write  $\frac{1}{2}$ , while I myself prefer writing `$1\over 2$` to obtain the same (note my use of parentheses – or rather, not using them where I need not)<sup>5</sup>. But once things get more complex, such as  $\frac{a}{\sqrt{1+q}}$ , the whole fraction increases in length, and lines get cramped or, even worse, the baseline stretch is increased. At any rate, if your equation is so complex, by all means revert to writing  $a/\sqrt{1+q}$ . Suggestion: don’t use fractions in inline equations, but use the simple  $/$  divider. What is easier to read,  $\frac{1}{2}$  or  $1/2$ ?

---

<sup>2</sup>Not quite true – depending on the kind of style you prefer. There are enough style guides preferring  $75\%$ , including the ISO 31-0 standard, while Knuth himself suggests a space, too. The Chicago Manual of Style disagrees. It’s up in the air, really!

<sup>3</sup>Here, too, no consensus.  $25^\circ\text{C}$  (Chicago style) and  $25^\circ\text{C}$  and  $25^\circ\text{C}$  are all acceptable – the latter not really in use anymore. No matter what: if you follow one style, be consistent.

<sup>4</sup>But, of course, these are only general conventions and not laws. Indeed, there exist equations where throwing in a `\cdot` to indicate scalar multiplication may actually clarify things.

<sup>5</sup>Preferring `\over` not the best choice; better get used to `\frac` as it is more robust to things like font specifications.

## 5.8 Vectorising

Sometimes it is important to clear the difference between a vector and a scalar. The standard way to indicate vectors is as  $\vec{x}$ , and that works fine if it is just one old vector lying around. But when a whole bunch is involved, for instance

$$\vec{x}\vec{y} + \vec{b}\vec{a} = 2,$$

the result looks silly. Some people prefer

$$\underline{x}\underline{y} + \underline{b}\underline{a} = 2$$

but, well, judge yourself.

One preferred way out is using boldface symbols. If you do, do it right. Get the  $\mathcal{A}\mathcal{M}\mathcal{S}$ math package and write,

```
\usepackage{amsmath}
\newcommand{\vect}[1]{\boldsymbol{#1}} % I would write, \def\vect#1{\boldsymbol{#1}}
```

to render  $\mathbf{abc}\phi\chi$ .

## 5.9 Collected math typesetting hints

- write `p(x \mid y)` to yield  $p(x \mid y)$  rather than `p(x|y)` which gives  $p(x|y)$ . But this goes wrong if the surrounding parentheses get larger: the `\mid` doesn't grow. In that case, you can use something like

```
\newcommand{\given}{\nonscript\,\middle\vert\,\nonscript\,}
$\p\left (x \given y \right)$
```

giving  $p(x \mid y)$ . But you then *must* use `\left...\right`, or else an error will occur. Be that as it may, my eye prefers the given spacing over what `\mid` does.

- write `\|x\|` to yield  $\|x\|$  rather than `||x||` which gives  $||x||$
- Easy to extend to, e.g., KL. But another problem arises:

```
\DeclareMathOperator{\KL}[2]{\KL\left(#1\nonscript\,\middle\|\,\nonscript\,,#2\right)}
$\KL{p(x)}{q(x)}$
```

rendering  $\text{KL}(p(x) \parallel q(x))$ . There is ugly space after the KL; and only limited control over the size of the parentheses.

Loading the package `mathtools` gives a solution, which we can exploit as follows:

```
\DeclarePairedDelimiterX{\KLx}[2]{\{ \} }{\#1\,,\delimsize\|\,,#2}
\newcommand{\KL}{\mathrm{KL}\KLx}
$\KL{p(x)}{q(x)}$ and $\KL[\big]{p(x)}{q(x)}$
```

rendering  $\text{KL}(p(x) \parallel q(x))$  and  $\text{KL}\left(p(x) \parallel q(x)\right)$ .

rendering  $\text{KL}(p(x) \parallel q(x))$  and  $\text{KL}\left(p(x) \Big\parallel q(x)\right)$ .



- sometimes you need to help: cf.

- `\exp(-(x-y) / (2\pi) )` giving  $\exp(-(x-y)/(2\pi))$

- `\exp\left(-(x-y) / (2\pi) \right)` giving  $\exp(-(x-y)/(2\pi))$

- `\exp\bigl(-(x-y) / (2\pi) \bigr)` giving  $\exp(-(x-y)/(2\pi))$

of which the last one is arguably easier to read.

## 6 Fonts

The topic of choosing fonts really goes beyond the scope of this short piece of text. But a nice short guide on typography is here. Basically: use serif fonts for long lines, as in a flow text. Basically: you're not at all bad off using Donald Knuth Computer Modern fonts.