

The L^AT_EX mini-guide

Alexandre Quenon

31st July 2018

Contents

I	L^AT_EX for beginners	1
1	Mathematics – Basics with the “mathtools” package	3
1.1	Packages for mathematics	3
1.2	Writing an equation	3
1.2.1	Unnumbered equations	4
1.2.2	Inline equations	5
1.2.3	Text-mode VS math-mode	5
1.2.4	Consequences of text- and math-mode: display style .	6
1.2.5	General recommendations for equations	6
1.3	Writing groups of equations	6
1.3.1	Group of equations	7
1.3.2	Group of aligned equations	7
1.3.3	General recommendations for groups of equations . . .	8
1.4	Common symbols for mathematics	8
II	L^AT_EX for common users	9
III	L^AT_EX for advanced users	11

Part I

L^AT_EX for beginners

Chapter 1

Mathematics – Basics with the “mathtools” package

Mathematics writing is one of the most advantage of L^AT_EX compared to common text editors. On a first approach, it looks like a programming language but it is in fact quite intuitive.

1.1 Packages for mathematics

The first package which was extremely useful in mathematics writing was *amsmath*. Since then, it has been upgraded by the *mathtools* package that I recommend to use. Hence the following line in the preamble of the document:

```
\usepackage{mathtools}
```

Other interesting packages are:

- *cases* which provides the `numcases` command to number all lines of a system of equations,
- *systeme* which provides command to format a system of equations for better readability, and
- *physics* which provides many commands to facilitate the writing of “complex” equations including derivatives and partial derivatives.

However, *systeme* is quite new at the time this document is being written and so it could lack of maturity.

1.2 Writing an equation

Usually, when a writer wants to put mathematics in a document, it takes the form of an equation. Writing an equation is simply done thanks to the

equation environment. Maxwell’s equations will be used as examples: so, the

```
\begin{equation}
  \vec{\nabla}\cdot\vec{B}=0.
  \label{eq::Maxwell:no_magnetic_monopole}
\end{equation}
```

L^AT_EX code generates

$$\vec{\nabla} \cdot \vec{B} = 0. \quad (1.1)$$

Here is the power of L^AT_EX: automatic numbering of equations. In *book*-like documents, equations are numbered within chapters by default, i.e., (1.x) in chapter 1, (2.x) in chapter 2 and so on. Moreover, equations numbers (a.k.a. tags) are placed on the right side. Of course, this layout can be modified.

1.2.1 Unnumbered equations

Automatic numbering can be avoided by using the starred version of the previous environment: `equation*`. For instance,

```
\begin{equation*}
  \vec{\nabla}\times\vec{E}
  =-\frac{\partial\vec{B}}{\partial t}.
\end{equation*}
```

produces

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}.$$

Nevertheless, it is generally recommended to number all equations in scientific documents for easier reference.

Shorter forms of the unnumbered version are offered by the package: the `\[... \]` wrapper or the double `$$` symbol. Please note that the latter is plain T_EX, which means that it should not be used with L^AT_EX because it is not robust. Though, the reason for which `$$` is presented in this document is that it is overly used on the Internet ¹.

Here follow the corresponding examples:

```
$$\iint_{\Sigma_f}\vec{B}\cdot\mathrm{d}\vec{S},$$
```

is the code corresponding to

$$\iint_{\Sigma_f} \vec{B} \cdot \mathrm{d}\vec{S},$$

while

1. And, as we all know, if it is on the Internet, it must be true...


```
\[ \oint_C \vec{E} \cdot \mathrm{d}\vec{l}
= - \iint_S \frac{\partial \vec{B}}{\partial t} \cdot \mathrm{d}\vec{S} . \]
```

creates

$$\oint_C \vec{E} \cdot d\vec{l} = - \iint_S \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S}.$$

However, I do not recommend to use short forms. On the one hand, the `equation*` environment highlights the mathematics when looking in the `LATEX` code. On the other hand, if the author changes his mind and wants to number the equation, he must simply remove the `*` character.

1.2.2 Inline equations

Inline equations are equations written in the text. It can be useful in some circumstances, such as the description of a variable. For instance, I could specify that \vec{B} in eq. (1.1) is the magnetic field. To do so, the equation is surrounded by single `$` signs:

```
$\vec{B}$
```

1.2.3 Text-mode VS math-mode

Inline equations underlines a fundamental behaviour of `LATEX`: the difference between *math-mode* and *text-mode*. As the names explain, text-mode is the regular mode of `LATEX` as the main part of the document is usually text, while math-mode is set in specific environments intended for mathematics.

Compare

```
regular behaviour (text-mode),
textinmathematicalenvironment(math-mode).
```

that is produced by

```
regular behaviour (text-mode),
$text in mathematical environment (math-mode)$.
```

Math-mode has several effects:

- a math font is used instead of the text font,
- the default font family is slanted while it is normal roman font in text,
- it manages white spaces in a different way than the ongoing typographical rules
 - any white space is automatically removed,
 - white space is added around mathematical operators,
- commands available in math-mode only will not generate errors.

1.2.4 Consequences of text- and math-mode: display style

Sometimes, inline equations may introduce unpleasant distortion in the text, especially with “big” symbols. As an example, let us use express the acceleration as the derivative of the speed: $a = \frac{dv}{dt}$. It can be seen that the fraction symbol has been compacted to fit with the line space. It happens when math-mode is used inside a block text-mode, like it is the case with inline equations.

It is possible to prevent the fraction from being reshaped by forcing the *displayed math-mode*. To do so, the writer must use the `displaystyle` command, which exists in shorter forms for common mathematical symbols such as fractions. Expressing again the acceleration: $a = \frac{dv}{dt}$, or, equivalently, $a = \frac{dv}{dt}$. It can be seen that the space line is increased above and below the line including the equation, creating a somewhat uncomfortable text arrangement.

The problem presented here above is related to the following L^AT_EX codes:

```
$a=\frac{\mathrm{d}v}{\mathrm{d}t}$
$a=\dfrac{\mathrm{d}v}{\mathrm{d}t}$
$\displaystyle a=\frac{\mathrm{d}v}{\mathrm{d}t}$
```

It will appear every time a “big” symbol (fraction, integral, sum, etc.) is used. However, it never occurs within mathematical environments such as `equation` because in that case the maths are *displayed*, i.e., horizontally centred and wrapped with additional vertical space in order to be highlighted and readable.

1.2.5 General recommendations for equations

Recommendations:

- use numbered equation only (with the `equation` environment),
- try to avoid inline equations except
 - to describe variables or operators,
 - for very small, less important and/or very well-known formulae which do not contain “big” symbols (e.g., integral, fraction).

1.3 Writing groups of equations

Several commands and environments allow to group equations. The most-used are presented here after. For a complete presentation, please refer to the *amsmath* and *mathtools* packages documentation.

1.3.1 Group of equations

The first tool which allow to group equations is the `gather` environment. Inside the environment, a double backslash (`\``\`) indicates the end of an equation and the beginning of a new one. Consequently, a new line is produced and another equation can be written. Pay attention: no double backslash must be put after the last equation. Otherwise, an additional space is added at the end of the group.

Unless the starred version (`gather*`) is used, all equations are numbered. To prevent one line from being numbered, the `\notag` or the `\nonumber` command can be used.

It is also possible to write text between equations while still being in the `gather` environment. This is done with the `intertext` command, or `shortintertext` which removes extra vertical space. It is specifically useful for a mathematical development. Within these commands, \LaTeX is in text-mode, while it is in math-mode within the rest of the `gather` environment.

As an example, the

```
\begin{gather}
\vec{\nabla} \times \vec{B} = \mu_0 \vec{j} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t},
\\
\int_C \vec{B} \cdot d\vec{l} = \mu_0 \iint_S \vec{j} \cdot d\vec{S} + \varepsilon_0 \mu_0 \iint_S \frac{\partial \vec{E}}{\partial t} \cdot d\vec{S}
\end{gather}
```

code will generate

$$\vec{\nabla} \times \vec{B} = \mu_0 \vec{j} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t}, \quad (1.2)$$

which can be written in the integral form by applying the Green theorem

$$\oint_C \vec{B} \cdot d\vec{l} = \mu_0 \iint_S \vec{j} \cdot d\vec{S} + \varepsilon_0 \mu_0 \iint_S \frac{\partial \vec{E}}{\partial t} \cdot d\vec{S} \quad (1.3)$$

(see the use of `\intertext`).

1.3.2 Group of aligned equations

The second tool allowing to group equations is the `align` environment. It does the same as the `gather` environment but also allows to align the equations. The alignment is performed thanks to the ampersand (&) symbol. All other commands and symbols performs the same as in the `gather` environment.

For instance,

```

\begin{align}
\vec{B} &= \vec{\nabla} \times \vec{A} \\
\vec{E} &= -\vec{\nabla} V - \frac{\partial \vec{A}}{\partial t}.
\end{align}

```

creates

$$\vec{B} = \vec{\nabla} \times \vec{A} \quad (1.4)$$

$$\vec{E} = -\vec{\nabla} V - \frac{\partial \vec{A}}{\partial t}. \quad (1.5)$$

1.3.3 General recommendations for groups of equations

Recommendations:

- use numbered equations only (unstarred versions) with possible exceptions for
 - numerical computations,
 - proofs (theorems, formulae, etc.),
- do not abuse of intertext because it makes the L^AT_EX code less readable.

1.4 Common symbols for mathematics

Mathematics would not be mathematics without any symbols. Common ones are:

- arithmetic operators such as + and −,
- comparison operators like > and <.

Less intuitive but still useful symbols are:

- the multiplication operator · generated by the `cdot` command,
- the fraction symbol $\frac{n}{d}$ created by the `frac` command,
- the integral \int which is produced by the `int` command, and
- any text symbol that is usually generated by a command having the same name (e.g., `sin` with the `sin` command or `exp` with `exp`).

There are many other symbols and being exhaustive is not the purpose of this mini-guide. If the reader is looking for specific symbols, he may refer to:

- his L^AT_EX editor, which generally provides a list of shortcuts and buttons to generate the correct commands,
- a quick review of symbols from ShareLaTeX,
- a big list of symbols native from T_EX and coming from different packages.

Part II

L^AT_EX for common users

Part III

L^AT_EX for advanced users

