The LaTeX mini-guide

Alexandre Quenon

1st August 2018

Contents

I LaTeX for beginners **1**

1 Mathematics – Basics **3**
1.1 Packages for mathematics 3
1.2 Writing equations 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 LaTeX for common users **9**

2 Mathematics – Matrices **11**
2.1 Packages for matrices 11
2.2 Writing matrices 11
2.3 Facilities for specific matrices 12
2.3.1 Matrices with specific patterns 12
2.3.2 Combinations of patterns 16

III LaTeX for advanced users **19**

Part I

LaTeX for beginners

Chapter 1

Mathematics – Basics

Mathematics writing is one of the most advantage of LaTeX 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 readility, 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 equations

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}
```

LaTeX code generates

$$\vec{\nabla}\cdot\vec{B}=0. \tag{1.1}$$

Here is the power of LaTeX: 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 $$ symbol. Please note that the latter is plain TeX, which means that it should not be used with LaTeX because it is not robust. Though, the reason for which $$ 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

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

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

creates

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

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

$$\text{regular behaviour (text-mode)},$$
$$textinmathematicalenvironment(math-mode).$$

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: $a = \frac{\mathrm{d}v}{\mathrm{d}t}$. 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: $a = \dfrac{\mathrm{d}v}{\mathrm{d}t}$, or, equivalently, $a = \dfrac{\mathrm{d}v}{\mathrm{d}t}$. 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 LaTeX 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{\jmath}
        + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t},
        \\
    \intertext{which can be written in the integral form by
        applying the Green theorem}
    \oint_{C} \vec{B} \cdot \mathrm{d} \vec{l}
        = \mu_0 \iint_S \vec{\jmath} \cdot \mathrm{d} \vec{S}
        + \varepsilon_0 \mu_0 \iint_S \frac{\partial \vec{E}}{
            \partial t} \cdot \mathrm{d} \vec{S}
\end{gather}
```

code will generate

$$\vec{\nabla} \times \vec{B} = \mu_0 \vec{\jmath} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t}, \tag{1.2}$$

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

$$\oint_C \vec{B} \cdot \mathrm{d}\vec{l} = \mu_0 \iint_S \vec{\jmath} \cdot \mathrm{d}\vec{S} + \varepsilon_0 \mu_0 \iint_S \frac{\partial \vec{E}}{\partial t} \cdot \mathrm{d}\vec{S} \tag{1.3}$$

(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

$$\vec{B} = \vec{\nabla} \times \vec{A} \tag{1.4}$$

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

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 LaTeX 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 $\cdot$ 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 LaTeX 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 TeX and coming from different packages.

Part II

LaTeX for common users

Chapter 2

Mathematics – Matrices

Engineers and scientists use many matrices to describe physical phenomena. Hence the frequently asked question: "how can I write a matrix with LaTeX"? This chapter answer the question.

2.1 Packages for matrices

As for the mathematics basics (cf. chapter 1), the *mathtools* package, which is an upgrade of the well-known and widely used *amsmath* package, provides everything that is necessary to write matrix computations.

In addition, the *physics* package provides very useful commands to facilitate the creation of specific matrices such as an identity matrix or a zero matrix.

So the following lines should be written in the preamble of the document:

```
\usepackage{mathtools}
\usepackage{physics}
```

2.2 Writing matrices

Matrices can be generated thanks to the `matrix` environment which must be used inside a mathematical equation environment. The simplest LaTeX code which generates a matrix is

```
\begin{equation*}
    \begin{matrix}
        x_{11} & x_{12} \\
        x_{21} & x_{22}
    \end{matrix}
\end{equation*}
```

and results in

$$\begin{matrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{matrix}$$

There are several variants of `matrix` which produce different delimiters surrounding the matrix. They are presented here below, with the corresponding environment's name on top of each matrix:

`matrix`
$$\begin{matrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{matrix}$$

`pmatrix`
$$\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}$$

`bmatrix`
$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$$

`Bmatrix`
$$\begin{Bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{Bmatrix}$$

`vmatrix`
$$\begin{vmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{vmatrix}$$

`Vmatrix`
$$\begin{Vmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{Vmatrix}$$

The *mathtools* package offers starred versions of the `matrix` environments which allow to pass an optional argument to specify the alignment within the matrix's columns. A LaTeX example is shown here below:

```
\begin{align*}
    \begin{pmatrix}
        2  & -3 \\
        42 & 0
    \end{pmatrix}    && \text{VS} &&
    \begin{pmatrix*}[r]
        2  & -3 \\
        42 & 0
    \end{pmatrix*}
\end{align*}.
```

Observe the difference in alignment between `pmatrix` (left) and `pmatrix*` (right):

$$\begin{pmatrix} 2 & -3 \\ 42 & 0 \end{pmatrix} \qquad \text{VS} \qquad \begin{pmatrix*}[r] 2 & -3 \\ 42 & 0 \end{pmatrix*}.$$

2.3 Facilities for specific matrices

The *physics* package proposes many commands that make the LaTeX's life of scientists and engineers easier. Among them, there are commands aiming the easy generation of specific matrices.

2.3.1 Matrices with specific patterns

In mathematics, there are specific matrices that are commonly used. These matrices are specific because they matches a pattern that is immediately recognised. The most known are the zero matrix, the identity matrix and the diagonal matrix.

The *physics* package provides commands to automatically generate the desired pattern for an $n \times m$ matrix, n and m being adapted according to the arguments passed to the commands. All commands must be placed inside a `matrix`-like environment, itself place within an `equation`-like environment.

Zero matrix

The zero matrix is generated with the `zeromatrix` command, or the shorter form `zmat`. The first argument is the dimension n, second argument is the dimension m of the $n \times m$ matrix. If m is omitted, the command generates a square zero matrix.

A comparison of the equivalent LaTeX codes

```
\begin{align*}
    % 3x3 matrix
    \begin{pmatrix} % based on physics
        \zmat{3}
    \end{pmatrix}   &&
    \begin{pmatrix} % regular way
        0 & 0 & 0 \\
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{pmatrix}   && \text{and} &&
    % 3x5 matrix
    \begin{pmatrix} % based on physics
        \zmat{3}{5}
    \end{pmatrix}   &&
    \begin{pmatrix} % regular way
        0 & 0 & 0 & 0 & 0 \\
        0 & 0 & 0 & 0 & 0 \\
        0 & 0 & 0 & 0 & 0
    \end{pmatrix}
\end{align*}
```

generating

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

shows that using the *physics* package avoids a fastidious copy-paste work.

Identity matrix

The identity matrix is generated with the `identitymatrix` command, or the shorter form `imat`. The argument is the dimension n of the square matrix.

Comparing the "regular" LaTeX with the code based on *physics*

```
\begin{equation*}
    \begin{pmatrix} % based on physics
        \imat{3}
    \end{pmatrix}   \quad
    \begin{pmatrix} % regular way
        1 & 0 & 0 \\
        0 & 1 & 0 \\
        0 & 0 & 1
    \end{pmatrix}
\end{equation*}
```

while both result in the same display

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

highligthts the time saved in this case, especially because the copy-paste is not as efficient as with the zero matrix.

Diagonal matrix

A diagonal matrix is generated with the `diagonalmatrix` command, or the shorter form `dmat`. The argument is a list of comma-separated values, the first one being x_{11}, the second one being x_{22}, and so forth. By default, only the values on the diagonal are printed and the rest is filled with white spaces. An optional argument allows to fill the spaces.

For instance, the following code

```
\begin{align*}
    % based on physics
    \begin{pmatrix}
        \dmat{a,b,c}
    \end{pmatrix}   &&
    \begin{pmatrix}
        \dmat{1,2,3}
    \end{pmatrix}   &&
    \begin{pmatrix}
        \dmat[0]{1,2,3}
    \end{pmatrix}   \\
    % regular way
    \begin{pmatrix}
        a &   &   \\
          & b &   \\
          &   & c
    \end{pmatrix}   &&
    \begin{pmatrix}
        1 &   &   \\
          & 2 &   \\
          &   & 3
    \end{pmatrix}   &&
    \begin{pmatrix}
        1 & 0 & 0 \\
        0 & 2 & 0 \\
        0 & 0 & 3
    \end{pmatrix}
\end{align*}
```

generates

$$\begin{pmatrix} a & & \\ & b & \\ & & c \end{pmatrix} \qquad \begin{pmatrix} 1 & & \\ & 2 & \\ & & 3 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix},$$
$$\begin{pmatrix} a & & \\ & b & \\ & & c \end{pmatrix} \qquad \begin{pmatrix} 1 & & \\ & 2 & \\ & & 3 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$

The above example also shows that literal values can be used as well as numerical values.

Similarly to the diagonal matrix, an anti-diagonal matrix can be created with the `antidiagonalmatrix` command, or the shorter form `admat`.

Generalised pattern

An $n \times m$ matrix can be filled with a specific element thanks to the `xmatrix` command, or the shorter form `xmat`. A starred version adds automatic indices.

As an example, the

```
\begin{align*}
    % based on physics
    \begin{pmatrix}
        \xmat{1}{2}{3}
    \end{pmatrix}   &&
    \begin{pmatrix}
        \xmat*{x}{3}{3}
    \end{pmatrix}   &&
    \begin{pmatrix}
        \xmat*{a}{3}{1}
    \end{pmatrix}   &&
    \begin{pmatrix}
        \xmat*{\alpha}{1}{3}
    \end{pmatrix}   \\
    % regular way
    \begin{pmatrix}
        1 & 1 & 1 \\
        1 & 1 & 1
    \end{pmatrix}   &&
    \begin{pmatrix}
        x_{11} & x_{12} & x_{13} \\
        x_{21} & x_{22} & x_{23} \\
        x_{31} & x_{32} & x_{33}
    \end{pmatrix}   &&
    \begin{pmatrix}
        a_{1} \\
        a_{2} \\
        a_{3}
    \end{pmatrix}   &&
    \begin{pmatrix}
```

```
        \alpha_{1} & \alpha_{2} & \alpha_{3}
    \end{pmatrix}
\end{align*}
```

LaTeX codes produces

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \quad \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \quad \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix}.$$

As observed, the automatic indexing takes the number of lines and columns into account. Automatic indexing is especially useful for general formulae and demonstrations. Finally, like for the diagonal matrix, literal (including Greek letters) values can be used.

At this point, the author believes that the reader is convinced by the power of the *physics* package to create matrices which match a specific pattern.

2.3.2 Combinations of patterns

Even though the aforementioned commands are usually sufficient for most cases, it is sometimes required to combine patterns together to obtain the desired result. The *physics* allow to do so [1].

A first possibility consists in creating the pattern and then adding the other elements around. The

```
\begin{equation*}
    \begin{pmatrix}
        \imat{2} \\
        a & b
    \end{pmatrix}    \quad
    \begin{pmatrix}
        a & b & c \\
        \zmat{2}{3}
    \end{pmatrix}
\end{equation*}
```

example code outputs

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{pmatrix} \quad \begin{pmatrix} a & b & c \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

However, if an element is added on the right or on the left of a pattern, the result is not the one expected, as depicted by the following code:

1. This section can be considered as intended for "advanced" users. However, the author thinks that separate it from the rest of the explanations about matrices is non-sense.

```
\begin{equation*}
    \begin{pmatrix}
        \imat{2} & a \\
                 & b
    \end{pmatrix}    \quad
    \begin{pmatrix}
                 & a \\
        \imat{2} & b
    \end{pmatrix}    \quad
    \begin{pmatrix}
        a & \\
        b & \zmat{2}{3}
    \end{pmatrix}.
\end{equation*}
```

$$\begin{pmatrix} 1 & 0 & \\ 0 & 1 & a \\ & b & \end{pmatrix} \quad \begin{pmatrix} & a & \\ 1 & 0 & \\ 0 & 1 & b \end{pmatrix} \quad \begin{pmatrix} a & & & \\ b & 0 & 0 & 0 \\ 0 & 0 & 0 & \end{pmatrix}.$$

To overcome this issue, the `matrixquantity` command, or the shorter form `mqty`, can be used. It creates a matrix as a single element.

Here follows an example that is explained below:

```
\begin{equation*}
    \begin{pmatrix}
        \mqty{\imat{2}} & \mqty{a\\b} \\
        \mqty{c & d}     & e
    \end{pmatrix}
\end{equation*}
```

$$\begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ c & d & e \end{pmatrix}.$$

Firstly, the identity matrix is created and embedded in `mqty` to be considered as a single element. Secondly, we can add a 2×1 vector on the right but he must also be encapsulated in `mqty` as the identity matrix is a single element. At this step, LaTeX sees a 1×2 matrix, even though the displayed result is a 2×3 matrix. Hence c and d that are embedded in `mqty` whereas e is a "regular" single element.

Part III

LaTeX for advanced users