

# L<sup>A</sup>T<sub>E</sub>X (& L<sup>Y</sup>X) Typesetting Tips

Version 1.3.4

Jörg C. Woehl

May 4, 2022

This document contains a collection of best practices using L<sup>A</sup>T<sub>E</sub>X (with or without L<sup>Y</sup>X). I have compiled them primarily for my own reference, but hope that you might find them useful, too.

Everything was thoroughly tested with the L<sup>A</sup>T<sub>E</sub>X packages mentioned below (all versions since May 2015) and L<sup>Y</sup>X 2.1.0 and later using XeTeX as the output engine.<sup>1</sup> LuaTeX seems to work without any problem as well. Each package was chosen after careful consideration of other options; I decided to add a package only if necessary, and only after I was convinced that it was stable, written by a reputable package author, playing nicely with the other packages, and under active development.

The latest version of this document along with the file `configJCW.sty` (see Section 3.6) and the accompanying L<sup>Y</sup>X macros file `configJCW.lyx` (see Section 7.2) are available at <https://github.com/JorgWoehl/LaTeXTips.git>.

## Contents

<b>1</b>	<b>Text</b>	<b>2</b>
1.1	Spaces after dots . . . . .	2
1.2	Spaces inside word groups . . . . .	3
1.3	Dashes . . . . .	3
<b>2</b>	<b>Numbers and units</b>	<b>4</b>
<b>3</b>	<b>Math</b>	<b>6</b>
3.1	Vectors, matrices, and tensors . . . . .	6
3.2	Mathematical constants and operators . . . . .	8
3.3	Extensible parentheses . . . . .	9

---

<sup>1</sup>XeTeX is enabled in L<sup>Y</sup>X by selecting PDF (XeTeX) as Default output format in Document ▷ Settings ▷ Formats. To make use of its extended font capabilities, Use non-TeX fonts (via XeTeX/LuaTeX) needs to be selected under Document ▷ Settings ▷ Fonts.

3.4	Multiplication sign . . . . .	10
3.5	Text in math mode . . . . .	11
3.6	Other L <sup>A</sup> T <sub>E</sub> X tips . . . . .	11
<b>4</b>	<b>Chemistry</b>	<b>12</b>
<b>5</b>	<b>Cross-references</b>	<b>13</b>
<b>6</b>	<b>Other packages</b>	<b>14</b>
6.1	microtype . . . . .	14
6.2	minted . . . . .	15
6.3	Add your custom package . . . . .	16
<b>7</b>	<b>L<sup>y</sup>X tricks</b>	<b>16</b>
7.1	L <sup>A</sup> T <sub>E</sub> X vs. LaTeX . . . . .	16
7.2	Displaying math macros in L <sup>y</sup> X . . . . .	16
7.3	Hyperref options . . . . .	17
7.4	Preventing package option clashes . . . . .	18
7.5	Adding modules . . . . .	18

# 1 Text

## 1.1 Spaces after dots

Between initials that are followed by a dot use a *thin space* `\,` (Option-ShiftSpace in L<sup>y</sup>X). After the last initial use an *interword space*<sup>2</sup> `\_` (Option-Command-Space<sup>3</sup>) – without adding any additional space after it. If the abbreviation is followed by a non-space character (such as a comma), the interword space is omitted.

J. C. Woehl	J. C. Woehl
J. C. Woehl	J.\,C.\ Woehl
Smith <i>et al.</i> show	Smith <code>\textit{et al.}</code> show
Smith <i>et al.</i> show	Smith <code>\textit{et al.}\</code> show

**Common abbreviations** such as *e.g.* or *i.e.* are typeset *without any space* between the letters, and are followed by an interword space (unless followed by a punctuation mark):

<sup>2</sup>An interword space is also sometimes called a control space or normal space.

<sup>3</sup>On macOS, you may need to deselect the option using the same shortcut (System Preferences > Keyboard > Shortcuts > Spotlight > Show Finder search window).

<i>e.g.</i> the following	<code>\textit{e.g.}</code> the following
<i>e.g.</i> the following	<code>\textit{e.g.}\</code> the following

## 1.2 Spaces inside word groups

Use a *protected (or non-breaking) space* `~` (`Option-Space`), *without* any space around it, **in word groups** such as references to named parts of a document,<sup>4</sup> between forenames, between multiple surnames, or wherever words are so closely linked that they should not be separated by a line break.

Samuel L. Jackson	<code>Samuel~L.~ Jackson</code>
-------------------	---------------------------------

A thin space is – by definition – also non-breaking, so `S.\,L.\ Jackson` is preferred over `S.~L.\ Jackson`.

Johannes Diderik van der Waals	<code>Johannes~Diderik van~der~Waals</code>
--------------------------------	---

Figure 3	<code>Figure~3</code>
----------	-----------------------

Note that typing out a cross-reference to a figure (or other part of the document), like in the preceding example, should never be necessary; this is better handled by the package `cleveref` (see Section 5).

## 1.3 Dashes

A *single dash* is used for hyphens in a multi-compound word:

my e-mail to a co-worker	<code>my e-mail to a co-worker</code>
--------------------------	---------------------------------------

A *double dash* (also called en-dash because it matches the width of the letter *n*) is used for number ranges or in text:

the July–August issues	<code>the July--August issues</code>
------------------------	--------------------------------------

pages 3–17	<code>pages~3--17</code>
------------	--------------------------

---

<sup>4</sup>The package `cleveref` takes care of cross-references to labels in a document.

the San Francisco–New York flight

the San Francisco--New York flight

A *triple dash* (also called em-dash because it matches the width of an uppercase *M*) is used to set off a parenthetical statement inside a sentence. The en-dash can also be used for this purpose if it is surrounded by spaces (which I personally prefer):

A flock of sparrows—some of them juveniles—alighted and sang.

A flock of sparrows---some of them juveniles---alighted and sang.

A flock of sparrows – some of them juveniles – alighted and sang.

A flock of sparrows -- some of them juveniles -- alighted and sang.

## 2 Numbers and units

The package `siunitx` takes care of the correct typesetting and formatting of numbers, units, and physical quantities that involve both. It is loaded by entering

```
\usepackage{siunitx}
```

(in Document▷Settings▷LaTeX Preamble when using L<sup>A</sup>T<sub>E</sub>X).

`siunitx` commands can be used both in text mode or math mode. The examples given below use the command names of version 3 of `siunitx`, which was released on September 29, 2021.<sup>5</sup>

When working in L<sup>A</sup>T<sub>E</sub>X, these commands are entered using Insert▷TeX code (Command-L) while in text mode,<sup>6</sup> or directly in math mode. For example, to enter a number followed by a unit, type `\qty{` followed by the numerical value, move the cursor outside the closing curly bracket, and enter `\{` followed by the unit symbol(s).

**Numbers** are entered using the `\num` command and are automatically spaced in groups of three (separated by a thin space) and without commas in the final output:

15 374.553 32

```
\num{15374.55332}
```

6.022 141 3 × 10<sup>23</sup>

```
\num{6.0221413e23}
```

<sup>5</sup>Old command names such as `\SI` and `\si` still remain available but are not recommended for use in new documents.

<sup>6</sup>Raw L<sup>A</sup>T<sub>E</sub>X commands inserted into the body of a L<sup>A</sup>T<sub>E</sub>X document are also often referred to as ERT (for evil red text), a term coined on the developers mailing list by Larry Marso in 2000.

**Angles** are entered using the `\ang` command; the angle sign is followed by the number without space:

12.3°	<code>\ang{12.3}</code>
-------	-------------------------

**Units** are entered using the `\unit` command in either an abbreviated format (where a dot signifies multiplication) or a more explicit textual format. `siunitx` automatically inserts a thin space between the unit symbols:

$\text{kg m s}^{-2}$	<code>\unit{kg.m.s^{-2}}</code>
$\text{kV}^2 \mu\text{m}^{-1}$	<code>\unit{\square\kilo\volt\per\um}</code>
°C	<code>\unit{\degreeCelsius}</code>

If you receive an `undefined control sequence` error when using the `\unit` command, you are probably experiencing a clash with the `unit` package, which also uses a command with this name (`siunitx` omits the definition of `\unit` if it detects the presence of the `unit` package). Avoid using the `\nicefrac` and `\unitfrac` commands, for example, as they [implicitly load](#) the `units` package<sup>7</sup>. In case of an unresolved package clash, you can revert back to using `\si` for units, which was the notation used in version 2 of `siunitx` and is still supported in version 3.

**Numbers followed by units** are entered using `\qty` following the same rules as above. `siunitx` automatically inserts a thin space between the number and unit:

$c = 299\,792\,458 \text{ m/s}$	<code>\$c=\qty{299792458}{m/s}\$</code>
20 °C	<code>\qty{20}{\degreeCelsius}</code>
$(0.1204 \pm 0.0007) \text{ nm}$	<code>\sisetup{separate-uncertainty} \qty{0.1204(7)}{\nm}</code>
760 mmHg	<code>\qty{760}{\mmHg}</code>
$8.314 \text{ J mol}^{-1} \text{ K}^{-1}$	<code>\qty{8.314}{\joule\per\mole\per\kelvin}</code>

<sup>7</sup>If you need fractions with a slanted fraction bar, you should use `\sfrac` from the `xfrac` package instead.

A vast array of possibilities for switches that alter the output format without changing the input (including rounding) is built into `siunitx`. In addition, lists and ranges of numbers as well as products and quotients of numbers can be entered directly, and numbers or units can be color-coded.

Because  $\text{LyX}$  does not natively support `siunitx`, only the raw  $\text{\LaTeX}$  code is displayed in  $\text{LyX}$ , which makes reading longer expressions difficult. This display issue can be resolved by defining corresponding  $\text{LyX}$  math macros for `siunitx` commands, as described in Section 7.2.

## 3 Math

The International Organization for Standardization (ISO) has developed a set of conventions for typesetting math in the physical and applied sciences. These recommendations are in agreement with the conventions specified in the red book of the International Union of Pure and Applied Physics (IUPAP) and the green book of the International Union for Pure and Applied Chemistry (IUPAC). The implementation of these recommendations in  $\text{\LaTeX}$  is described below.

In what follows it is assumed that the `mathtools` package is loaded, which is an extension of the `amsmath` package. In  $\text{LyX}$ , it can be enabled by choosing Document▷Settings▷Math Options▷mathtools and selecting Load always, or by loading it in the  $\text{LaTeX}$  Preamble with:

```
\usepackage{mathtools}
```

### 3.1 Vectors, matrices, and tensors

Unfortunately, even with the `mathtools` package there are certain ISO math typesetting conventions that  $\text{\LaTeX}$  is unable to cover. For example, physical quantities should be typeset in *italic*, vectors and matrices in ***bold italic***, tensors in ***sans-serif bold italic***, while mathematical operators and constants (see Section 3.2) should be typeset in upright font.  $\text{\LaTeX}$  supports these scenarios for Latin characters, but typesets lowercase Greek variables only in italic and uppercase Greek variables only in upright font. The `amsmath` package provides some uppercase Greek characters in italic (e.g. `\varDelta` for  $\Delta$ ), but support for boldface Greek symbols is inconsistent.

The `unicode-math` package provides a fix for these font issues. It can be loaded in the  $\text{LaTeX}$  Preamble:<sup>8</sup>

---

<sup>8</sup>Alternatively, the `unicode-math` package can be loaded in  $\text{LyX}$  by selecting Non- $\text{\TeX}$  Fonts Default in the Math field in Document▷Settings▷Fonts (the box Use non- $\text{\TeX}$  fonts (via  $\text{XeTeX}$ / $\text{LuaTeX}$ ) needs to be checked as well). The option `math-style=ISO` is passed to `unicode-math` by entering it in the Custom field under Document Class, or entered in the  $\text{LaTeX}$  Preamble with `\unimathsetup{math-style=ISO}`.

```
\usepackage[math-style=ISO]{unicode-math}
```

`unicode-math` uses the Latin Modern Math font by default. Other fonts can be loaded by adding a line to the LaTeX Preamble, such as:

```
\setmathfont{xits-math.otf}
```

A list of currently supported fonts is available on the `unicode-math` [homepage](#).

$bD\beta\Delta$	<code>\$\mathbf{D} \mathit{\beta} \Delta\$</code>
$\mathbf{D}\beta\Delta$	<code>\$\mathbf{\mathrm{D}} \mathit{\beta} \Delta\$</code>
$\mathbf{D}\beta\Delta$	<code>\$\mathbf{\mathrm{D}} \mathit{\beta} \Delta\$</code>
$\mathbf{D}\beta\Delta$	<code>\$\mathbf{\mathrm{D}} \mathit{\beta} \Delta\$</code>
$\mathbf{D}\beta\Delta$	<code>\$\mathbf{\mathrm{D}} \mathit{\beta} \Delta\$</code>

For consistency, it is suggested to define the following semantic markup commands in the LaTeX Preamble,

```
\newcommand*\mathbf[1]{\mathbf{#1}}
\newcommand*\mathit[1]{\mathit{#1}}
\newcommand*\mathsf[1]{\mathsf{#1}}
```

and use these new commands instead of directly changing font specifications.<sup>9</sup> Vectors can also be entered using a top arrow (`\vec`) in LyX; they can be automatically converted to the new notation by adding the following to the LaTeX Preamble:

```
\AtBeginDocument{
  \renewcommand*\vec{\mathbf}
}
```

---

If the `unicode-math` package is loaded with the `math-style=ISO` option in the LaTeX Preamble, and the Math field in Document > Settings > Fonts is set to Non-TeX Fonts Default without enabling the same option, an option clash error will occur (see Section 7.4 for more information about this type of error).

<sup>9</sup>These commands could additionally be wrapped in `\ensuremath` to use the markup could in both text mode and math mode. Using `\{\}\ensuremath`, however, is controversial practice, and does not have a real benefit for entering commands in LyX – it is just as easy to press Command-L to enter the macro in text mode as it is to press Command-M to enter it in math mode. In addition, display issues in LyX due to unsupported macros can only be addressed in math mode (Section 7.2).

The `\AtBeginDocument` command ensures that the redefinition happens at the beginning of the document so that other packages cannot tamper with it.

vector $\boldsymbol{E}$ or $\boldsymbol{E}$	vector <code><math>\vec{E}</math></code> or <code><math>\vect{E}</math></code>
vector $\boldsymbol{\mu}$	vector <code><math>\vec{\mu}</math></code>
matrix $\boldsymbol{A}$	matrix <code><math>\matr{A}</math></code>
tensor $\boldsymbol{A}$	tensor <code><math>\tens{A}</math></code>

Although the fonts may not be properly displayed in LyX itself, they will appear correctly in the final output. Note that display issues in LyX can be overcome by defining (protected) LyX math macros as described in Section 7.2.

Also note that `unicode-math` cannot be used with pdfLaTeX, which is the default TeX engine in modern L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> distributions; however, it runs fine with XeTeX or LuaTeX, which are also included in these distributions.

## 3.2 Mathematical constants and operators

*Mathematical* constants (such as Euler’s number  $e$ , the imaginary unit  $i$ , or the number  $\pi$ ) and mathematical operators (such as the differential operator  $d$ , the partial differential operator  $\partial$ , or the difference operator  $\Delta$ ) are typeset in upright font. L<sup>A</sup>T<sub>E</sub>X respects this convention for operators that are entered as commands:

$\sin x$	<code><math>\sin x</math></code>
$\sin x$	<code><math>\sin x</math></code>

Also, some special symbols such as the `\nabla`  $\nabla$  operator are typeset correctly, but this is not the case for other mathematical constants and operators.

The best way to deal with this situation is to define the semantic markup `\const{}` and use shortcuts like `\ee` or `\ii` for frequently used mathematical constants:

```
\newcommand*\const{[1]{\symup{#1}}}
\newcommand{\ee}{\const{e}}
\newcommand{\ii}{\const{i}}
```

The Greek letter  $\pi$  almost always refers to the mathematical constant  $\pi$  and can therefore simply be redefined globally (at the beginning of the document). In order to avoid an infinite recursive loop, the old definition must be assigned to a new command first:

```
\AtBeginDocument{
  \let\oldpi\pi
  \renewcommand{\pi}{\const{\oldpi}}
}
```



Note that the two lines inside the `\AtBeginDocument` environment can simply be added to an existing `\AtBeginDocument` environment.

For differentials and difference operators we proceed similarly:

```
\newcommand*{\oper}[1]{\mathop{}\!\symup{#1}}
\newcommand{\dd}{\oper{d}}
\newcommand{\pd}{\oper{\partial}}
\newcommand{\Dd}{\oper{\Delta}}
```

The empty `\mathop` command and negative space `\!` is used to provide the correct spacing before the operator signs. This definition distinguishes  $dx$  from  $d$  times  $x$  and automatically leads to the correct spacing in all contexts in which these operators may be used (which makes it, for example, unnecessary to insert *thin spaces* `\,` before differentials when used in math mode).

$e^{i\pi} + 1 = 0$	<code>\$\ee^{\ii\pi} + 1 = 0\$</code>
$f(x) dx$	<code>\$f(x)\dd x\$</code>

### 3.3 Extensible parentheses

If a mathematical expression is enclosed in parentheses, use *extensible* parentheses to accommodate for different vertical space needs. In  $\text{\LaTeX}$ , extensible parentheses are entered by clicking on  $()$  or typing **Control-M**  $()$ .

Extensible parentheses should *always* be used around values for physical quantities (except when they appear alone) to make the equations easier to read:

$\frac{3}{2} (8.314 \text{ J K}^{-1} \text{ mol}^{-1})$	<code>\$\frac{3}{2} \left( \right. \backslash\text{qty}\{8.314\}\{J.K^{-1}.mol^{-1}\} \left. \right)\$</code>
---	---

However,  $\text{\TeX}$  adds additional space around extensible parentheses, which is not always welcome:

$f(x)$	<code>\$f \left( x \right)\$</code>
$f(x)$	<code>\$f ( x )\$</code>

Therefore, **avoid** using extensible parentheses when they enclose the **argument of a function or operator**, such as in the function  $f(x)$ , differential  $d(xy)$ , derivative  $\frac{d}{dx}(xy)$ ,

or difference  $\Delta(xy)$ . If extensible parentheses are necessary to accommodate the argument, insert a *negative thin space* `\!` before them to cancel the extra space (in L<sup>A</sup>T<sub>E</sub>X, just type `\!` or use the space popup in the math toolbar).

$d(\sum x_i)$	<code>\displaystyle</code> <code>\dd ( \sum x_i )\$</code>
$d(\sum x_i)$	<code>\displaystyle</code> <code>\dd \left( \sum x_i \right)\$</code>
$d(\sum x_i)$	<code>\displaystyle</code> <code>\dd \! \left( \sum x_i \right)\$</code>

### 3.4 Multiplication sign

The multiplication operator  $\times$  (type `\times` in L<sup>A</sup>T<sub>E</sub>X) should be used for all multiplications involving *numerical values* in order to avoid potential confusion with the decimal point. The package `siunitx` (Section 2) does this automatically when multi-part numbers or quantities are entered.

$3 \times 4 \times 8$	<code>\numproduct{3 x 4 x 8}\$</code>
$3 \times 10^{10}$	<code>\num{3e10}\$</code>
$3 \text{ cm} \times 4 \text{ cm} \times 8 \text{ cm}$	<code>\qtyproduct{3 x 4 x 8}{cm}\$</code>

Everywhere else the centered dot  $\cdot$  (type `\cdot` in L<sup>A</sup>T<sub>E</sub>X) can be used if so desired, such as between parentheses or between symbolic quantities.

$(3 \text{ cm}) \cdot (5 \text{ m s}^{-2})$	<code>\left( \qty{3}{cm} \right) \cdot</code> <code>\left( \qty{5}{m.s^{-2}} \right)\$</code>
$a \cdot b$	<code>a \cdot b\$</code>

### 3.5 Text in math mode

Text in math mode that appears in a *subscript or superscript* is entered using the `\textnormal` command,<sup>10</sup> However, it is recommended to use the `\text` command for *all other text* in math mode, because it adapts to the surrounding text environment by using the same font. Instead of entering the command, you can use **Command-M** to toggle text mode on and off while within math mode. Both commands require `amsmath` (which is automatically loaded by the `mathtools` package).

$V_{\text{tot}}$	<code>\$V_{\textnormal{tot}}\$</code>
$pV = nRT$ (ideal gas)	<code>\$pV=nRT \quad \text{(ideal gas)}\$</code>

Subscripts and superscripts are typeset using `\textnormal` if they are **descriptive** (representing the name of a person or a particle), but remain italic if they represent a **variable or quantity**:

Boltzmann's constant, $k_{\text{B}}$ (name of a person)
mass of an electron, $m_{\text{e}}$ (name of a particle)
heat capacity at constant pressure, $C_p$ (quantity)

### 3.6 Other L<sup>A</sup>T<sub>E</sub>X tips

#### Eqnarray

Avoid using the `eqnarray` environment, which is **very buggy**. Use the AMS `align` environment instead (Insert ▷ Math ▷ AMS align Environment in L<sup>A</sup>T<sub>E</sub>X).

#### Avogadro's Number

Insert a **negative space** `\!` into the *subscript* of Avogadro's number  $N_{\text{A}}$  (if the negative space is put directly after the  $N$ , it has the same effect on the subscript, but potential superscripts will now be drawn too close to it):

<sup>10</sup>This is preferred over `\mathrm` or `\textrm`. `\mathrm` uses math roman font, but uses math spacing (ignoring spaces) and interprets dashes as minus signs, while `\textrm` uses roman font that adjusts to the text environment (which would lead to italic sub/superscripts in a theorem environment).

$N_A$	<code>\$N_\textnormal{A}\$</code>
$N_A$	<code>\$N_\textnormal{\!A}\$</code>

## Definitions

To typeset an equation definition, define the following command in the LaTeX Preamble:

```
\newcommand{\eqdef}{\overset{\textnormal{def}}{=}}
```

$a \stackrel{\text{def}}{=} b$	<code>\$a \eqdef b\$</code>
--------------------------------	-----------------------------

## LaTeX Preamble

In order to avoid a messy preamble, put configuration-related commands in a file `config.sty` and call it from the LaTeX Preamble with

```
\usepackage{config}
```

Follow the instructions in Section 6.3 to add this package to your TeX system.

A file named `configJCW.sty` with the recommendations contained in this document is available at <https://github.com/JorgWoehl/LaTeXTips.git>.

## 4 Chemistry

Typesetting chemical formulas and reaction equations is done with the package `mhchem`.

Although this package can be selected directly using Document > Settings > Math Options in LyX, it loads by default with the older version 3 format and without any options. As there seems to be no way to change the options of packages loaded natively by LyX, it is suggested to select **Do not load** (to avoid a package option clash; see Section 7.4) and use the following code in Document > Settings > LaTeX Preamble instead (which also uses prettier arrows rather than those from the current math font<sup>11</sup>):

```
\usepackage[version=4,arrows=pgf-filled]{mhchem}
```

<sup>11</sup>The `layout` option for stacked superscripts and subscripts is not recognized by `mhchem` v3.21, even though it is officially referenced in the documentation. This is working correctly in v4.08. (Note that the `layout` option should be avoided anyhow as it does not conform to IUPAC conventions.)

The `mhchem` commands are entered in L<sup>A</sup>T<sub>E</sub>X as TeX code (`Command-L`) when in text mode; they can be typed in directly in math mode.

$\text{SO}_4^{2-} + \text{Ba}^{2+} \longrightarrow \text{BaSO}_4 \downarrow$	<code>\ce{S04^2- + Ba^2+ -&gt; BaS04 v}</code>
$A \xrightleftharpoons{x_i} B$	<code>\ce{A &lt;--&gt;[\$x_i\$] B}</code>
$\text{CO}_2(\text{g}) + \text{C}(\text{s}) \xrightleftharpoons[\text{below}]{k_1, \Delta} 2 \text{CO}(\text{g})$	<code>\ce{C02(g) + C(s) &lt;=&gt;[\$k_1, \Delta\$][below] 2C0(g)}</code>
$\text{Hg}^{2+} \xrightarrow[\text{red}]{\text{I}^-} \text{HgI}_2$	<code>\ce{Hg^2+ -&gt;[\ce{I-}] \$ \underset{\mathrm{red}}{\ce{HgI2}} \$}</code>
${}^{227}_{90}\text{Th}^+$	<code>\ce{^{227}_{90}Th+}</code>
$\text{RNO}_2 \xrightleftharpoons{+e} \text{RNO}_2^{\bullet -}$	<code>\ce{RN02 &lt;=&gt;[+e] RN02^{\bullet -}}</code>
$V_{\text{H}_2\text{O}}$	<code>\$V_{\ce{H2O}}\$</code>
$A \cdots B = C = D \equiv E$	<code>\ce{A\bond{...}B\bond{~~}C=D#E}</code>
$a \text{A} + b \text{B} \longrightarrow c \text{C} \quad \Delta_{\text{f}} H^\circ < 0$	<code>\ce{\$a\$A + \$b\$B -&gt; \$c\$C \quad \Delta_{\textnormal{f}} H^\circ &lt; 0}</code>

## 5 Cross-references

Cross-references to figures, tables, and parts of a document are best done using the package `cleveref`. It automatically determines the type of cross-reference and the context in which it is used, and can even typeset lists of cross-references.

`cleveref` is designed to work with the `hyperref` package (which turns cross-references into clickable hyperlinks in the pdf output), but `cleveref` must be loaded last. The documentation even states that `cleveref` should be loaded *after* all other packages that don't specifically support it.

```
\usepackage[noabbrev,capitalize]{cleveref}
```

This loads the package with options that prevent abbreviations like eq. or fig. and capitalizes all occurrences (Equation, Figure). The hyperref package can be loaded in LyX by checking Use Hyperref Support in Document ▷ Settings ▷ PDF Properties; it can also be loaded using

```
\usepackage{hyperref}
```

in the LaTeX Preamble (as long as it is loaded before cleveref).

LyX normally uses prettyref or refstyle for cross-references, and does not natively support cleveref. Fortunately, there is a simple workaround: enter all cross-references as usual using LyX’s built-in functionality by selecting the format <reference>, and convert these \ref commands into \cref commands during the compile phase (but before hyperref goes to work). This can be achieved by adding the following lines to the end of the LaTeX Preamble (right after the cleveref package is loaded):

```
\AtBeginDocument{
  \renewcommand{\ref}{\cref}
}
```

Section 1

```
\ref{sec:Typesetting-text}
```

Footnote 4

```
\ref{fn:Cross-references-to-labels}
```

Cross-references containing more than one label are entered as TeX code (Command-L):

```
\crefrange{fig:PoissA}{fig:PoissB}
```

```
\cref{fig:PoissA,fig:PoissE}
```

## 6 Other packages

### 6.1 microtype

The `microtype` package provides “subliminal refinements towards typographical perfection”. To use it in LyX, check Enable micro-typographic extensions in Document ▷ Settings ▷ Fonts, or load it in the LaTeX Preamble:

```
\usepackage{microtype}
```

## 6.2 minted

The `minted` package provides syntax highlighting of source code listings in  $\text{\LaTeX}$  using the `Pygments` library. To use it, the source file must be compiled with the `-shellescape` flag. Starting with  $\text{LyX}$  2.3.0, this is done by simply checking **Allow running external programs** in **Document**  $\triangleright$  **Settings**  $\triangleright$  **Formats** for the source file in question (this setting is not sticky; even if it is set in a template file, it has to be enabled again for all child documents). The first time the source file is compiled, a warning dialog will be displayed, where the option is confirmed by clicking the **Always allow for this document** button.

In older versions of  $\text{LyX}$ , the compile option has to be enabled globally, which can be achieved by selecting the proper converter in **LyX**  $\triangleright$  **Preferences**  $\triangleright$  **File Handling**  $\triangleright$  **Converters** and adding the `-shell-escape` flag to the Converter string, e.g.

```
xelatex -shell-escape $$i
```

Click on **Modify** and **Save** to apply the change. The converter in use can be determined by clicking on **View Complete Log** when a  $\text{\TeX}$  error is generated after `minted` is used for the first time. For example, if the log starts with “This is XeTeX” you have to add the shell-escape flag to the `xelatex` converter (click on the **LaTeX (XeTeX) -> PDF (XeTeX)** entry).

`minted` requires `Pygments`, which can be installed using the Python Package Index on macOS. At the command line prompt, type

```
pip install Pygments
```

Alternatively, `Pygments` can also be installed using `MacPorts`. At the command line prompt, type

```
port search pygments
```

and choose the `pygments` version with the highest number (*e.g.* `py34-pygments`). Install it with

```
sudo port install py34-pygments
```

No matter how `Pygments` is installed, you will need to put a soft link to `pygmentize` (use `which -a pygmentize` to get the path to the version you want to use) inside the `/usr/local/bin` folder, so that  $\text{\TeX}$  can find it:

```
sudo ln -s /path/to/pygmentize /usr/local/bin/pygmentize
```

## 6.3 Add your custom package

Write your own package using the following format:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{mypackage}
% Your new commands here.
% Load packages with
% \RequirePackage[options]{package}
% instead of
% \usepackage[options]{package}
\endinput
```

If you are on macOS, put a symbolic link in the folder `~/Library/texmf/tex/latex/` (create the `texmf` folder and all subfolders if they do not yet exist) that points to your package file:

```
sudo ln -s /path/to/mypackage.sty ~/Library/texmf/tex/latex/mypackage.sty
```

Follow a similar procedure if you are on Windows or Linux.

Note: Before defining a new command, look up the command name in the [The Comprehensive LaTeX Symbol List](#) to make sure that it is still available.

## 7 LyX tricks

### 7.1 $\LaTeX$ vs. LaTeX

Starting with LyX 2.2, “LaTeX” is rendered as normal text, while the  $\LaTeX$  logo is inserted using [Insert ▷ Special Character ▷ Logos ▷ LaTeX Logo](#). Before version 2.2, LyX rendered the text “LaTeX” automatically as  $\LaTeX$ ; to prevent this from happening, put it inside a  $\TeX$  code box ([Command-L](#)).

### 7.2 Displaying math macros in LyX

To [display math macros in LyX that it does not natively support](#) (for example those that were added through packages or in your own [LaTeX Preamble](#)), insert a comment at the beginning of your document ([Insert ▷ Note ▷ Comment](#)). Inside this comment, enter one or more LyX math macros ([Insert ▷ Math ▷ Macro](#)). Putting LyX math macros inside a comment allows their use in the LyX document, while protecting them from being processed by  $\LaTeX$  for the final output. This makes it possible to provide two definitions of a given  $\LaTeX$  macro that will not interfere with each other: one in the



L<sup>A</sup>T<sub>E</sub>X preamble or inside a package for the final output, and a second one for use with the instant preview feature in L<sup>A</sup>T<sub>E</sub>X (the instant preview renderer does not recognize packages defined in the L<sup>A</sup>T<sub>E</sub>X preamble). Instant preview can be toggled on and off in Preferences ▷ Display.

A L<sup>A</sup>T<sub>E</sub>X macro has three fields labeled Name, TeX, and LyX. Arguments are entered in the TeX and LyX fields by typing `\#1` (with a backslash) for argument 1, and so on; there is no need to enter them in the Name field as they are automatically added to it.

- In the Name field, enter the name of the math macro you want to add support for (*e.g.* the vector macro `\vect` defined earlier). Do not type the backslash when entering the macro, as it is automatically added.
- In the TeX field, provide the code that will be used for instant preview (here: `\symbf{\#1}`, entered by typing `\symbf\{` followed by `\#1`).<sup>12</sup> Note that you can leave the TeX field empty if you are certain that you won't use instant preview; should you decide to use it later on, however, your macros would magically disappear in displayed math formulas (although they would still be present in the final output)!
- Finally, in the LyX field, use a graphical representation that mimics how the macro will look like in the final output. This could be the same command as in the TeX field if L<sup>A</sup>T<sub>E</sub>X can render it, in which case the LyX field can simply be left empty. In our example, `\symbf` is not rendered by L<sup>A</sup>T<sub>E</sub>X. To mimic the final output, we could make the argument `\#1` boldface (Edit ▷ Math ▷ Text Style ▷ Math Bold Series). However, I prefer to decorate `\#1` with a small arrow instead, which is a more recognizable vector notation in normal L<sup>A</sup>T<sub>E</sub>X formulas.

A L<sup>A</sup>T<sub>E</sub>X macros file `configJCW.lyx` for all commands recommended in this document is available at <https://github.com/JorgWoehl/LaTeXTips.git>.

In order to use math macros in L<sup>A</sup>T<sub>E</sub>X, simply type the L<sup>A</sup>T<sub>E</sub>X command in math mode and hit the spacebar. If the macro contains arguments, one or more boxes will appear for entering them. The cursor will automatically be placed in the first argument box; to jump to the next argument box, use the right arrow or tab key.

## 7.3 Hyperref options

Options for the `hyperref` package (such as `allcolors=blue`, which changes the default color for all types of links to blue) can be added by entering them in the Additional options field in Document ▷ Settings ▷ PDF Properties. In the Hyperlinks tab, you should check the options No frames around links and Color links.

---

<sup>12</sup>If your code contains commands that are not recognized by the instant preview rendering engine, only the usual L<sup>A</sup>T<sub>E</sub>X math mode will be displayed.

## 7.4 Preventing package option clashes

Package option clashes occur when packages are loaded a second time with options that are *not a subset* of the options passed in the first package loading call (in other words, when the second package loading call is more specific than the first one). A package option clash always produces an error during compilation of the  $\text{\LaTeX}$  document. Therefore, while it is safe to load packages explicitly (in the  $\text{\LaTeX}$  preamble or via a package style file) or implicitly (as part of other packages) *without* specifying options, care must be taken when packages are loaded *with options*. If these same options are not part of the first package loading call, an error will occur.

A particular source of frustration is that it is not always evident which packages LyX is loading.<sup>13</sup> For example, selecting **Non-TeX Fonts Default** in the **Math** field in **Document**  $\triangleright$  **Settings**  $\triangleright$  **Fonts** when the box **Use non-TeX fonts (via XeTeX/LuaTeX)** is checked loads the `unicode-math` package – which is far from obvious! However, it is easy to find out which packages are loaded, and which options are passed to it, by taking a look at the  $\text{\LaTeX}$  code that LyX produces (**View**  $\triangleright$  **Code Preview Pane**, selecting **Preamble Only**). Keep **Automatic update** selected; this is handy for finding out which (if any) packages are loaded in response to certain selections in LyX’s user interface or when certain  $\text{\LaTeX}$  commands are invoked<sup>14</sup>.

## 7.5 Adding modules

To install a module in LyX, you must put it in the `layouts` subdirectory of the LyX configuration directory, which is listed in the dialog box that opens when clicking **LyX**  $\triangleright$  **About LyX** or **Help**  $\triangleright$  **About LyX**. Creating a symbolic link to it works as well:

```
sudo ln -s /path/to/mymodule.module ~/path/to/LyX/layouts
```

Then run **Tools**  $\triangleright$  **Reconfigure** to let LyX know about it. After restarting LyX, the new module should now be available under the list of available modules (**Document**  $\triangleright$  **Settings**  $\triangleright$  **Modules**).

Note that major releases of LyX are installed in their own subdirectories; the preceding procedure thus needs to be repeated if a new major LyX release is installed.

---

<sup>13</sup>LyX loads packages that it natively supports *before* any packages defined in the  $\text{\LaTeX}$  preamble.

<sup>14</sup>For example, as mentioned in Section 2, using the commands `\nicefrac` or `\unitfrac` in the document loads the `units` package, which can lead to a package clash with `siunitx`.