

LyX's detailed Math manual

by the LyX Team*

Version 1.6.x

August 31, 2010

*If you have comments or error corrections, please send them to the LyX Documentation mailing list: lyx-docs@lists.lyx.org

Contents

1. Introduction	1
2. General Instructions	1
3. Basic Functions	4
3.1. Exponents and Indices	4
3.2. Fractions	4
3.3. Roots	5
3.4. Binomial Coefficients	6
3.5. Case Differentiations	6
3.6. Negations	7
3.7. Placeholders	7
3.8. Lines	8
3.9. Ellipses	8
4. Matrices	9
5. Brackets and Delimiters	11
5.1. Vertical Brackets and Delimiters	11
5.1.1. Manual Bracket Size	11
5.1.2. Automatic Bracket Size	12
5.2. Horizontal Brackets	13
6. Arrows	14
6.1. Horizontal Arrows	15
6.2. Vertical and diagonal Arrows	15
7. Accents	16
7.1. Accents for one Character	16
7.2. Accents for Operators	16
7.3. Accents for several Characters	17
8. Space	18
8.1. Predefined Space	18
8.2. Variable Space	19
8.3. Space besides inline Formulas	19
9. Boxes and Frames	20
9.1. Boxes with Frame	20
9.2. Boxes without Frame	21
9.3. Colored Boxes	22
9.4. Paragraph Boxes	23

10. Operators	25
10.1. Big Operators	25
10.2. Operator Limits	26
10.3. Binary Operators	28
10.4. Self-defined Operators	28
11. Fonts	29
11.1. Font Styles	29
11.2. Bold Formulas	30
11.3. Colored Formulas	31
11.4. Font Sizes	31
12. Greek Letters	33
12.1. Small Letters	33
12.2. Big Letters	33
12.3. Bold Letters	33
13. Symbols	34
13.1. Mathematical Symbols	34
13.2. Miscellaneous Symbols	34
13.3. The Euro-Symbol €	35
14. Relations	35
15. Functions	36
15.1. Predefined Functions	36
15.2. Self-defined Functions	36
15.3. Limits	37
15.4. Modulo-Functions	38
16. Special Characters	38
16.1. Special Characters in Mathematical Text	38
16.2. Accents in Text	38
16.3. Minuscule Numbers	39
16.4. Miscellaneous special Characters	40
17. Formula Styles	40
18. Multiline Formulas	41
18.1. General	41
18.1.1. Line Separation	41
18.1.2. Column Separation	42
18.1.3. Long Formulas	42
18.1.4. Multiline Brackets	43
18.2. Align Environments	44

18.2.1. Standard align Environment	44
18.2.2. Alignat Environment	45
18.2.3. Flalign Environment	45
18.3. Eqnarray Environment	45
18.4. Gather Environment	46
18.5. Multline Environment	46
18.6. Multiline Formula Parts	47
18.7. Text in multiline Formulas	47
19. Formula Numbering	48
19.1. General	48
19.2. Cross-References	48
19.3. Subnumbering	49
19.4. User-defined Numbering	50
19.5. Numbering with Roman Numbers and Letters	51
20. Chemical Symbols and Equations	52
21. Diagrams	54
21.1. Amscd Diagrams	54
21.2. Xymatrix Diagrams	55
22. User-defined Commands	55
22.1. The Command <code>\newcommand</code>	55
22.2. Math Macros	56
23. Tips	60
23.1. Negative Numbers	60
23.2. Comma as decimal Separator	60
23.3. Physical Vectors	60
23.4. Self-defined Fractions	61
23.5. Canceled Formulas	62
23.6. Formulas in Section Headings	62
23.6.1. Heading without formula in table of contents	63
23.6.2. Heading with formula in table of contents $\sqrt{-1} = i$	63
23.7. Formulas in multi-column Text	63
23.8. Formulas with Description of Variables	64
23.9. Upright small Greek Letters	65
23.10. Text Characters in Formulas	65
A. Typographic Advice	67
B. Synonyms	68
References	69

1. Introduction


This document explains L^AT_EX's math features and is furthermore a collection of L^AT_EX-commands used for mathematical characters and constructs. The explanations are designed for the usage of commands. It is therefore required that you have read the section `Mathematical Formulas` of the User's Guide.

Most of the characters and many constructs explained in this manual are also accessible via the menu `Insert`▷`Math`, or the `math toolbar`. But everybody who has to write lots of formulas will notice that it is much faster to use commands instead of the `math toolbar`. Therefore this manual is focused on commands but also mentions the corresponding toolbar buttons when available.

If not specially mentioned the commands are only available within formulas. To be able to use all commands explained in this document, the option `Use AMS math package` must be used in the document settings (menu `Document`▷`Settings`▷`Math Options`).¹

This document doesn't list all $\mathcal{A}\mathcal{M}\mathcal{S}$ -math commands² for lucidity reasons.

2. General Instructions

To create an inline formula that is embedded into a text line, use one of the shortcuts `Ctrl+M`, `Alt+C M`, `Alt+M M` or the toolbar button 

To create a display style formula that will appear bigger and in an own paragraph, use one of these shortcuts: `Ctrl+Shift+M`, `Alt+M D`.

To change a display style formula to an inline formula, set the cursor into the formula and use one of the shortcuts `Ctrl+M`, `Alt+C M`, `Alt+M M` or the menu `Edit`▷`Math`▷`Change formula type`. The same way is used to change an inline formula to a display style formula.

To display parts of an inline formula in the size of a display style formula, enter the command `\displaystyle` to a formula. Then a new blue box appears in which the desired formula part is inserted.

Only inline formulas are allowed inside tables.

The `math toolbar` can be turned on in the menu `View`▷`Toolbars`. When you click there on "Math" the toolbar will be shown permanently at the bottom; this state is visualized in the `Toolbars` menu with a checkmark. When you click in this state again

¹The option `Use AMS math package` automatically only uses $\mathcal{A}\mathcal{M}\mathcal{S}$ -math when math constructs are found that are supported by L^AT_EX.

²A list with all $\mathcal{A}\mathcal{M}\mathcal{S}$ -math commands is in the file [amsguide.ps](#), which is part of every L^AT_EX standard installation.

on “Math” in the **Toolbars** menu, the **math toolbar** is only shown when the cursor is within a formula; this state is visualized by the renaming of the menu entry from “Math” to “Math (auto)”.

The \TeX -mode is invoked by pressing the toolbar button **TeX** or by using the menu **Insert**▷**TeX Code** (shortcut **Ctrl+L**).

To change the \LaTeX -preamble, use the menu **Document**▷**Settings**▷**LaTeX Preamble**.

To edit matrices, case differentiations and multiline formulas subsequently, the menus **Edit**▷**Math** and **Edit**▷**Rows & Columns**, or the **table toolbar** can be used. When lines and columns are swapped via the menu, the column or line where the cursor is in is exchanged with the column to the right or the line below, respectively. Is the cursor in the last column or row, the exchange is done with the column to the left or the line above.

To write text in formulas³ *mathematical text* is used. This mode is invoked with the shortcut **Alt+C Space** or by the insertion of the command `\text`. The text appears black in \LyX and can therefore be distinguished from the other formula parts that appear blue. In the output mathematical text is set upright, in contrary to other formula parts.

Command Scheme

Most of the \LaTeX -commands for math constructs have the following scheme:

`\commandname[optional argument]{required argument}`

A command starts always with a backslash „\“. To omit optional arguments, also omit the associated brackets. The braces around the required arguments are named in this document as \TeX -braces. If you add in a formula a left brace to a command name, \LyX creates automatically a \TeX -brace. In all other cases \TeX -braces are created in formulas with the command `\{`. \TeX -braces appear red in \LyX , in contrary to normal braces that appear blue. In \TeX -mode no command is needed to get \TeX -braces. \TeX -braces don’t appear in the output.

When commands without arguments, like commands for symbols are entered in \TeX -mode, a space character must *always* be behind the command to end it. This space doesn’t appear in the output. When the space should appear in the output, the space must be followed by a protected space in normal text.

A protected space is inserted with **Ctrl+Space**.

³For multiline formulas the command `\intertext` is used, see sec. 18.7.

Syntax Explanation

- The symbol⁴ \square denotes a space character to be input.
- An arrow like \rightarrow denotes the usage of the corresponding arrow key on the keyboard.

Available units



Table 1: Available units

Unit	Name / Description
mm	Millimeter
cm	Centimeter
in	Inch (1 in = 2,54 cm)
pt	Point (72.27 pt = 1 in)
pc	Pica (1 pc = 12 pt)
sp	scaled point (65536 sp = 1 pt)
bp	big point (72 bp = 1 in)
dd	Didot (1 dd \approx 0.376 mm)
cc	Cicero (1 cc = 12 dd)
ex	Height of letter “ <i>x</i> ” in the current font
em	width of letter “ <i>M</i> ” in the current font
mu	math unit (1 mu = $\frac{1}{18}$ em)

⁴This visible space character can be created with the command `\textvisiblespace`, inserted in T_EX-mode.

3. Basic Functions



3.1. Exponents and Indices

Indices are created with an underscore “_” or via the math toolbar button , exponents with a caret “^” or via the math toolbar button .

command	Result
<code>B_V</code>	B_V
<code>B^V</code>	B^V
<code>B^_A</code>	B^A

As the caret is in some languages an accent, vowels will be accentuated in this case and not set as exponents⁵. To get in this case exponents, press **Space** after the caret as in the last example.

3.2. Fractions

Fractions are generated with the command `\frac` or via the math toolbar button . The font size is adjusted automatically, depending on whether the fraction is in an inline or display style formula. With the math toolbar button  you can select different fraction types.

With the command `\dfrac` a fraction can be created that has in any case the size of a display style formula. With `\tfrac` the fraction appears always with the size of an inline formula. An example:

A line with the fraction $\frac{1}{2}$ that was created with the command `\frac`.

A line with the fraction $\frac{1}{2}$ that was created with the command `\dfrac`.

Command	Result
<code>\frac_A B</code>	$\frac{A}{B}$
<code>\dfrac_A B</code>	$\frac{A}{B}$
<code>\dfrac_e^{\frac{1}{2}}_3</code>	$e^{\frac{1}{2}}_3$

⁵Depending on the used keyboard settings this can also happen for other characters than vowels.

For nested fractions the command `\cfrac` can be used. Here is an example:

created with `\frac`

$$\frac{A}{B + \frac{C + \frac{E}{F}}{D}}$$

created with `\cfrac`

$$\cfrac{A}{B + \cfrac{C + \frac{E}{F}}{D}}$$

The command for the example above is:

`\cfrac{A}{B + \cfrac{C + \cfrac{E}{F}}{D}}`

`\cfrac` sets the fraction always in the size of a displayed formula, also when it is part of another fraction.

It is possible to specify the alignment of the numerator. The command `\cfracleft` is used to left align it, the command `\cfracright` to right-align it. `\cfrac` centers the numerator. These fractions demonstrate the different alignments:

$$\frac{A}{B + C}, \quad \frac{A}{B + C}, \quad \frac{A}{B + C}$$

Note: `\cfracleft` and `\cfracright` are no real \LaTeX commands but represent the command `\cfrac[alignment]{numerator}{denominator}`. Therefore you cannot use them in \TeX code.

It is often advantageous to combine `\cfrac` and `\frac`:

$$\frac{A}{B + \cfrac{C + \frac{E}{F}}{D}}$$

For inline fractions with a sloped fraction stroke you can use the command `\nicefrac`: $\frac{5}{31}$ or `\unitfrac`: $\frac{5}{31}$ There is furthermore the command `\unitfracthree` that offers to write a fraction in combination with a number: $2\frac{1}{3}$

Note: `\unitfracthree` is not a real \LaTeX command but the command `\unitfrac[number]{numerator}{denominator}`. Therefore you cannot use it in \TeX code.

How to define own fractions where the fraction stroke can be changed, is explained in sec. 23.4.

3.3. Roots

Square roots are created with `\sqrt` or the math toolbar button $\sqrt{\square}$, all other roots with the command `\root` or with the math toolbar button $\sqrt[n]{\square}$.

Command	Result
<code>\sqrt{A-B}</code>	$\sqrt{A-B}$
<code>\root{3}{A-B}</code>	$\sqrt[3]{A-B}$

A square root can also be created with `\root` when the root index field is left empty.


With certain indices the distance to the root is too small, like in this formula: $\sqrt[\beta]{B}$. The β touches the root. To avoid this, the commands `\leftroot` and `\uproot` are used with the following scheme:

`\leftroot{distance}` and `\uproot{distance}`

Distance is the number of Big Points (unit bp; 72 bp = 1 inch), that the index should be moved to the left or top, resp.. The commands are written to the index. This way the command

`\root\leftroot{-1}\uproot{2}\beta\rightarrow B`
 produces a correct typeset formula: $\sqrt[\beta]{B}$



3.4. Binomial Coefficients

Binomial coefficients are inserted with the command `\binom` or with the submenu of the math toolbar button . Analog to fractions (`\frac`) there are besides `\binom` the commands `\dbinom` and `\tbinom`. For other brackets around binomial coefficients there are the commands `\brack` and `\brack`.

Command	Result
<code>\binom{A}{B}</code>	$\binom{A}{B}$
<code>\dbinom{A}{B}</code>	$\dbinom{A}{B}$
<code>\tbinom{A}{B}</code>	$\tbinom{A}{B}$
<code>\brack{A}{B}</code>	$\brack{A}{B}$
<code>\brace{A}{B}</code>	$\brace{A}{B}$

3.5. Case Differentiations

Command	Result
<code>\cases{A \rightarrow B > 0}</code>	$\left\{ \begin{array}{l} A \\ B > 0 \end{array} \right.$
<code>\cases{Ctrl+Return}</code>	$\left\{ \begin{array}{l} A \text{ for } x > 0 \\ B \text{ for } x = 0 \end{array} \right.$

After inserting `\cases` or the usage of the math toolbar button  you can create new lines with the shortcut `Ctrl+Return` or the table toolbar button .

The command `\cases` is also available via the menu `Insert > Math > Cases-Environment`.

3.6. Negations

By inserting of `\not` every character can be displayed canceled. The characters are quasi accentuated by a slash.


Command	Result
<code>\not=</code>	\neq
<code>\not \le</code>	\nless
<code>\not \parallel</code>	\nparallel

The last example shows, that not all negations look good. Therefore there are for some negations special commands (see sec. 13.1 and sec. 14).

3.7. Placeholders

When displaying e. g. isotopes⁶ the following problem occurs:



Indices created with sub- and superscripts: ${}_{9}^{19}\text{F}$
 correct indices: ${}^{19}_{9}\text{F}$

The shorter index is by default placed below or above the first character of the longer index. To avoid this there is the command `\phantom` or the math toolbar button  that creates one or more phantom characters. When inserting `\phantom` a small blue box appears that is superposed with two red arrows. The arrows indicate that the complete width and height of the box content will be created as placeholder. Phantom characters are accordingly placeholders with the size of the characters.

Command	Result
<code>^19_{9}\text{F}</code>	${}^{19}_{9}\text{F}$
<code>^{235}_{93}\text{F}</code>	${}^{235}_{93}\text{F}$
<code>^t_{MM}\Lambda</code>	Λ_{MM}^t

⁶Typesetting isotopes and chemical symbols is described in sec. 20.

⁷can be found in the submenu of the toolbar button 

Furthermore there are the commands `\vphantom` (toolbar button ) and `\hphantom` (toolbar button ). `\hphantom` creates only space for the maximal height of the characters in the box but not for its width. `\vphantom` creates only space for the width of the box content. Therefore the boxes of both commands have only one red arrow.

For example `\vphantom_a\int` space for the height of the integral sign,⁸ because this is the larger character. An example application is in sec. 18.1.4.

3.8. Lines

Command	Result
<code>\overline{A+B}</code>	$\overline{A + B}$
<code>\underline{A+B}</code>	$\underline{A + B}$
<code>\overline{\underline{A+B}}</code>	$\overline{\underline{A + B}}$

In the last example it doesn't matter if first `\overline` or `\underline` is inserted.

To double underline e. g. results, one uses `\underline` twice.

It is possible to place up to 6 lines above or below characters.

Custom lines can be created using the command `\rule` which has the following scheme:

`\rule[vertical offset]{length}{thickness}`

The optional vertical offset shifts the line upwards (or downwards, when the value is negative). The units listed in Table 1 can be used for the values. Here are two example lines created with the commands

`\rule[-2ex]{3cm}{2pt}` and `\rule{2cm}{1pt}`:

This is a sentence  with two lines.

`\rule` can also be used for text when it is inserted in T_EX-mode.

3.9. Ellipses

There are different types of ellipses available.⁹ For listings dots at the baseline are used (`\ldots`), while for operations dots are needed that are on the same height as the operators (`\cdots`). When using the command `\dots`, L^AT_EX decides on the basis of the next character what type is used.

⁸The command `\int` creates an integral sign, see sec. 10.1.

⁹In the math toolbar in the submenu of the button \cdots

Command	Result
A_1, \dots, A_n	A_1, \dots, A_n
$A_1 + \dots + A_n$	$A_1 + \dots + A_n$
A_1, \ldots, A_n	A_1, \dots, A_n
$A_1 + \cdots + A_n$	$A_1 + \dots + A_n$
\vdots	\vdots
\ddots	\ddots
3×3 matrix with the different dots	$\begin{matrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{matrix}$

The ellipses available in menu **Insert**▷**Special Character** are $\backslash\mathbf{ldots}$.

Specially for matrices there are ellipses that span over several columns. They are created with the command $\backslash\mathbf{hdotsfor}$, that has the following scheme:

$\backslash\mathbf{hdotsfor}[\mathbf{distance}]\{\mathbf{number\ of\ columns}\}$

The number of columns specifies how many columns should be spanned. Distance is a factor for the distance between the dots.

In the following matrix the command $\backslash\mathbf{hdotsfor}[2]\{4\}$ was inserted in the first box of the second line, to get an ellipsis with a dot distance twice as long as with the command $\backslash\mathbf{dots}$:

$$\begin{pmatrix} A & B & C & D \\ \dots & \dots & \dots & \dots \\ q & w & e & r \end{pmatrix}$$

Note that the matrix fields that should be spanned must be empty, otherwise you get L^AT_EX-errors.

Furthermore you can fill with the command $\backslash\mathbf{dotfill}$ the rest of a line with dots. The effect of these commands is the same like with $\backslash\mathbf{hfill}$, see sec. 8.2.

For example the command $\mathbf{A}\backslash\mathbf{dotfill}\mathbf{B}$ produces


A.....B

Analog to $\backslash\mathbf{dotfill}$ there is for a line the command $\backslash\mathbf{hrulefill}$:

A_____B

To use the commands for text, they have to be inserted in T_EX-mode.

4. Matrices

Matrices can be inserted via the math toolbar button  or the menu **Insert**▷**Math**▷**Matrix**. You will be asked for the number of matrix columns and rows and the

alignment. The vertical alignment is hereby only of importance for matrices in inline formulas:

The first matrix is top aligned, the second middle aligned, and the third bottom aligned.



$$\begin{array}{cccc} A & D & G & J \\ B & E & H & K \\ C & F & I & L \end{array}$$

$$\begin{array}{cccc} A & D & G & J \\ B & E & H & K \\ C & F & I & L \end{array}$$

The horizontal alignment specifies how the column entries should be aligned. It is set by entering a letter for every column. *l* denotes left aligned, *c* centered and *r* right aligned. To create for example a 4×4 matrix where the first column is left aligned, the second and third are centered and the last one is right aligned, one enters for the horizontal alignment **lccr**. Normally are in a matrix all columns centered, therefore the default is for every column is a **c**.

Horizontal alignment:

lll : $\begin{array}{cccc} 10000 & D & G & \\ B & 10000 & H & \\ C & F & 10000 & \end{array}$, **ccc** : $\begin{array}{cccc} 10000 & D & G & \\ B & 10000 & H & \\ C & F & 10000 & \end{array}$, **rrr** : $\begin{array}{cccc} 10000 & D & G & \\ B & 10000 & H & \\ C & F & 10000 & \end{array}$

To add or delete rows and columns subsequently, the math toolbar buttons , , etc., or the menu **Edit**▷**Rows & Columns** can be used. New rows can also be created with **Ctrl+Return**.

Parentheses around a matrix can either be created with the commands **\left** and **\right** (shortcut **Alt+M** Parenthesis), see sec. 5.1.2, or by using the following commands:

Command	Result	Command	Result
<code>\bmatrix_2×2 matrix</code>	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	<code>\vmatrix_2×2 matrix</code>	$\begin{vmatrix} 0 & -i \\ i & 0 \end{vmatrix}$
<code>\Bmatrix_2×2 matrix</code>	$\begin{Bmatrix} 0 & -i \\ i & 0 \end{Bmatrix}$	<code>\Vmatrix_2×2 matrix</code>	$\begin{Vmatrix} 0 & -i \\ i & 0 \end{Vmatrix}$
<code>\pmatrix_2×2 matrix</code>	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	<code>\matrix_2×2 matrix</code>	$\begin{matrix} 0 & -i \\ i & 0 \end{matrix}$

When e. g. **\vmatrix** is inserted, a blue box appears between two vertical lines where the matrix is inserted.

As all multiline formulas are matrices, the length **\arraycolsep** that is described in sec. 18.1.2 can also be used to change the column separation of matrices.

To change the row separation, the command `\arraystretch` is used. It is used as follows:

`\renewcommand{\arraystretch}{stretch factor}`

The command `\renewcommand` assigns the stretch factor to the predefined command `\arraystretch`. To double e.g. the row separation, use the factor 2. This is then used for all following matrices. To go back to the original separation, assign the factor 1 to `\arraystretch`.

To set matrices into a text line, the command `\smallmatrix` is used. When it is inserted a blue box with two dashed lines appears. In this box the matrix is inserted.

This is a matrix $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ in a text line.

5. Brackets and Delimiters

5.1. Vertical Brackets and Delimiters

Command	Result	Command	Result
<code>(</code>	$($	<code>)</code>	$)$
<code>{</code>	${$	<code>}</code>	$}$
<code>[</code>	$[$	<code>]</code>	$]$
<code>\langle</code>	\langle	<code>\rangle</code>	\rangle
<code>\lceil</code>	\lceil	<code>\rceil</code>	\rceil
<code>\lfloor</code>	\lfloor	<code>\rfloor</code>	\rfloor
<code>/</code>	$/$	<code>\ </code>	$\ $
<code> </code>	$ $	<code>\ </code>	$\ $

Note: In T_EX-mode the command `\textbackslash` must be used for the backslash, because the command `\\` produces there a line break.

For all characters listed above the size can be adjusted with the commands described in the following two subsections. When using these commands, the characters `<` and `>` can directly be used instead of the commands `\langle` and `\rangle`.

5.1.1. Manual Bracket Size

The bracket size can be specified manually by the L^AT_EX-commands `\big`, `\Big`, `\bigg` and `\Bigg`. `\big` denotes the smallest and `\Bigg` the largest bracket size.

These commands are used to emphasize levels of brackets:

all brackets in the same size: $((A + B)(A - B))^C$

this looks better: $\left((A + B)(A - B)\right)^C$

For the second formula the command $\backslash\mathbf{Big}((\mathbf{A}+\mathbf{B})(\mathbf{A}-\mathbf{B})\backslash\mathbf{Big})^{\frown}\mathbf{C}$ has been used. Here is an overview about all bracket sizes:

$\backslash\mathbf{Bigg}(\backslash\exp\backslash\mathbf{bigg}\langle\backslash\mathbf{Big}[\backslash\mathbf{big}\{\ln(3x)\backslash\mathbf{big}\}^2\backslash\sin(x)\backslash\mathbf{Big}]^{\frown}\mathbf{A}\backslash\mathbf{bigg}\rangle\backslash\mathbf{Bigg})^{\frown}0,5$


$$\left(\exp\left\langle\left[\left\{\ln(3x)\right\}^2\sin(x)\right]^{\frown}\right\rangle\right)^{0,5}$$

Besides the $\backslash\mathbf{big}$ -commands there is the variant $\backslash\mathbf{bigm}$ that adds a bit more space between the bracket and its content, and the variant $\backslash\mathbf{bigl}\backslash\mathbf{bigr}$, that don't add additional space. The l at the end of the command $\backslash\mathbf{bigl}$ is for a left bracket; for a right bracket this will be replaced by an r . A left or right bracket can each be an opening or closing bracket.

In the following table is a comparison of the variants:

Command	Result
$\backslash\mathbf{Bigm}(\backslash\mathbf{bigm}(\backslash\ln(3x)\backslash\mathbf{bigm})^2\backslash\mathbf{Bigm})$	$\left(\left(\ln(3x)\right)^2\right)$
$\backslash\mathbf{Big}(\backslash\mathbf{big}(\backslash\ln(3x)\backslash\mathbf{big})^2\backslash\mathbf{Big})$	$\left(\left(\ln(3x)\right)^2\right)$
$\backslash\mathbf{Bigl}(\backslash\mathbf{bigl}(\backslash\ln(3x)\backslash\mathbf{bigr})^2\backslash\mathbf{Bigr})$	$\left(\left(\ln(3x)\right)^2\right)$
$\backslash\mathbf{bigl}\backslash\ln(3x)\backslash\mathbf{bigr}(\backslash\mathbf{bigl}\backslash\ln(3x)\backslash\mathbf{bigr})$	$\ln(3x)\left(\ln(3x)\right)$

5.1.2. Automatic Bracket Size

Brackets with variable size can be inserted with the commands $\backslash\mathbf{left}$ and $\backslash\mathbf{right}$ or via the math toolbar button . Directly behind $\backslash\mathbf{left}$ and $\backslash\mathbf{right}$ the wanted bracket must be inserted. The bracket size will then automatically be calculated for the output.

normal bracket: The command $\backslash\ln(\backslash\mathbf{frac}\mathbf{A}\downarrow\mathbf{C}\downarrow)$ creates

$$\ln\left(\frac{A}{C}\right)$$

multiline bracket: The command $\backslash\ln\backslash\mathbf{left}(\backslash\mathbf{frac}\mathbf{A}\downarrow\mathbf{C}\downarrow\backslash\mathbf{right})$ creates

$$\ln\left(\frac{A}{C}\right)$$

Instead of `\left` and `\right` the shortcut `Alt+M Bracket` can be used. This has the advantage that you can see in LyX immediately the real bracket size and that the matching right bracket will be created too.

The command for the last example would then be: `\ln Alt+M (\frac{A}{B}C`

To omit a left or right bracket, a dot is inserted for the omitted bracket. For example the command `\left.\frac{A}{B}\right\}` creates:

$$\left.\frac{A}{B}\right\}$$

The commands `\left` and `\right` will be converted by LyX to brackets in the right size when the document is reloaded and an omitted bracket will appear as dashed line.

Because all popular L^AT_EX-Distributions use e_TE_X, an extension to L^AT_EX, the command `\middle` is additionally available for all brackets and limits. With this command the height of the following character is adapted to the one of the surrounding brackets, what is e. g. needed for physical vectors:

$$\left\langle \phi \middle| J = \frac{3}{2}, M_J \right\rangle$$

For physical vectors there is a special L^AT_EX-package, described in sec. 23.3.

5.2. Horizontal Brackets

Command	Result
<code>\overbrace{A+B}^3</code>	$\overbrace{A+B}^3$
<code>\underbrace{A+B}_5</code>	$\underbrace{A+B}_5$
<code>\overbrace{\underbrace{A+B}_w}^C</code>	$\overbrace{\underbrace{A+B}_w}^C$

In the last example it doesn't matter if `\overbrace` or `\underbrace` is inserted at first.

When brackets are needed that overlap each other, multiline formulas, as described in sec. 18, must be used:

$$A = \underbrace{\overbrace{ggg + bbqq}^r + dddd}_s$$

In the first row the formula is inserted together with the first brace. It is hereby important that the space command¹⁰ `\:` is inserted before the first d , because the brace that ends behind the q prevents that the following “+” is surrounded by space.¹¹ In the second row the second brace is inserted. As it should begin before the b , first the command `\hphantom{gggg+\:}` is inserted.¹² This space is needed because the “+” is also surrounded by space in the formula. The brace is placed under the command `\hphantom{bbqq+dddd}`.

It gets more complicated when brackets overlap each other, like in the following example:

$$A = \underbrace{gggg + bbqq + dddd}_r$$

The first formula row is the same as the second row of the previous example, with the difference that the brace is above. The second row contains the formula together with the second brace. To avoid that there is space between the upper brace in the first row and the formula, the row spacing need to be reduced. This is not easily possible due to a bug in LyX¹³. As solution for the problem, the global formula row separation `\jot` must be changed to `-6pt` before the formula with the command `\setlength{\jot}{-6pt}` in TeX-mode. `\jot` is set back after the formula to the standard value of `3pt` using the same command. More about the row separation in formulas is explained in sec. 18.1.1.

6. Arrows

Arrows can be inserted via the math toolbar button  or the commands listed in the following subsections.

¹⁰Space commands are explained in sec. 8.1.

¹¹because a bracket is not handled as character, see sec. 10.3

¹²more about `\hphantom` see sec. 3.7

¹³[LyX-bug #1505](#)

6.1. Horizontal Arrows

Command	Result
<code>\gets</code>	\leftarrow
<code>\Leftarrow</code>	\Leftrightarrow
<code>\longleftarrow</code>	\longleftarrow
<code>\Longleftarrow</code>	\Longleftarrow
<code>\leftharpoonup</code>	\curvearrowleft
<code>\leftharpoondown</code>	\curvearrowright
<code>\hookleftarrow</code>	\hookleftarrow

Command	Result
<code>\leftrightarrow</code>	\leftrightarrow
<code>\Leftrightarrow</code>	\Leftrightarrow
<code>\longleftrightarrow</code>	\longleftrightarrow
<code>\Longleftrightarrow</code>	\Longleftrightarrow
<code>\rightleftharpoons</code>	\rightleftharpoons

Command	Result
<code>\to</code>	\rightarrow
<code>\Rightarrow</code>	\Rightarrow
<code>\longrightarrow</code>	\longrightarrow
<code>\Longrightarrow</code>	\Longrightarrow
<code>\rightharpoonup</code>	\curvearrowright
<code>\rightharpoondown</code>	\curvearrowleft
<code>\hookrightarrow</code>	\hookrightarrow

Command	Result
<code>\mapsto</code>	\mapsto
<code>\longmapsto</code>	\longmapsto
<code>\leadsto</code>	\leadsto
<code>\dashrightarrow</code>	\dashrightarrow

Arrows used as accent like e. g. vector arrows are listed in sec. 7.

Furthermore there are the labeled arrows `\xleftarrow` and `\xrightarrow`. When inserting one of these commands in a formula, an arrow with two blue boxes appear where the label can be inserted. The length of the arrow adapts to the label width.

Command	Result
$F(a)\xleftarrow[x>0]{x=a}F(x)$	$F(a) \xleftarrow[x>0]{x=a} F(x)$
$F(x)\xrightarrow[x>0]{x=a}F(a)$	$F(x) \xrightarrow[x>0]{x=a} F(a)$

6.2. Vertical and diagonal Arrows

Command	Result
<code>\uparrow</code>	\uparrow
<code>\Uparrow</code>	\Uparrow
<code>\updownarrow</code>	\updownarrow
<code>\Updownarrow</code>	\Updownarrow
<code>\Downarrow</code>	\Downarrow
<code>\downarrow</code>	\downarrow

Command	Result
<code>\nearrow</code>	\nearrow
<code>\searrow</code>	\searrow
<code>\swarrow</code>	\swarrow
<code>\nwarrow</code>	\nwarrow

Vertical arrows can be used also as delimiter together with the commands described in sec. 5.1.1 and sec. 5.1.2.

7. Accents

Accents can be inserted via the math toolbar button  or the commands listed in the following subsections.

7.1. Accents for one Character¹⁴

Command	Result
<code>\dot_A</code>	\dot{A}
<code>\ddot_A</code>	\ddot{A}
<code>\dddotted_A</code>	\dddot{A}
<code>\ddddotted_A</code>	\ddddot{A}
<code>\vec_A</code>	\vec{A}
<code>\bar_A</code>	\bar{A}
<code>\mathring_A</code>	\mathring{A}

Command	Result
<code>\tilde_A</code>	\tilde{A}
<code>\hat_A</code>	\hat{A}
<code>\check_A</code>	\check{A}
<code>\acute_A</code>	\acute{A}
<code>\grave_A</code>	\grave{A}
<code>\breve_A</code>	\breve{A}

You can directly insert accents like \acute{e} to formulas. LyX will transform them to the corresponding accent command. For umlauts it is better to insert a quotation mark before the vowel. These two characters are then treated by L^AT_EX as *one* character when the formula part with the umlaut is marked as German. In contrary to `\ddot`, with this method “real” umlauts are created as demonstrated in the following example:

Command	Result
<code>“i</code>	\ddot{i}
<code>\ddot_i</code>	\ddot{i}

Another advantage to `\ddot` is that umlauts can directly be converted to mathematical text because the accent commands above are *not allowed in mathematical text*. To convert an accented character to mathematical text, only the character under the accent may be converted. This applies also for all other conversions, e. g. to italic or bold.

In mathematical text, umlauts and other accented characters can directly be inserted.

7.2. Accents for Operators

With the commands `\overset` and `\underset` characters can be placed above or below an operator, respectively, to accent it. With the command `\sideset` characters can be set before and behind an operator. The command scheme is:

¹⁴accents in text see sec. 16.2

`\sideset{character before}{character behind}`

`\sideset` must always be before the operator that should be accented. You can accent with several characters and even with other operators and symbols. To place with `\sideset` for example only characters behind an operator, write nothing between the first braces but don't omit the braces.

For example the command `\sideset{\rightarrow\{\}'\rightarrow\sum_{k=1}^n` produces:

$$\sum_{k=1}^n$$

The command `\overset{\maltese}{a}` produces:



As seen in the last example, with `\overset` and `\underset` also symbols and characters can be accented; with `\sideset` this is not possible.

7.3. Accents for several Characters

Command	Result	Command	Result
<code>\overleftarrow{A=B}</code>	$\overleftarrow{A=B}$	<code>\overrightarrow{A=B}</code>	$\overrightarrow{A=B}$
<code>\underleftarrow{A=B}</code>	$\underleftarrow{A=B}$	<code>\underrightarrow{A=B}</code>	$\underrightarrow{A=B}$
<code>\overleftrightarrow{A=B}</code>	$\overleftrightarrow{A=B}$	<code>\widetilde{A=B}</code>	$\widetilde{A=B}$
<code>\underleftrightarrow{A=B}</code>	$\underleftrightarrow{A=B}$	<code>\widehat{A=B}</code>	$\widehat{A=B}$

With these commands as many characters as you like can be accented. But the accents `\widetilde` and `\widehat` will only be set in the output with a length of three characters, as shown in the following example:


$$A + \widetilde{B} = C - D$$

With the commands `\overset` and `\underset` described in the previous subsection it is also possible to accent several characters. The command `\underset{***}{A=B}` creates:

$$\underset{***}{A=B}$$

8. Space

8.1. Predefined Space

Sometimes it is necessary to insert horizontal space to a formula. This is done by inserting a protected space (shortcut **Ctrl+Space**). A “`␣`” appears and by pressing **Space** several times one can select one of eight different space sizes. The spaces can also be inserted using the math toolbar button  or special commands. Independent from the inserted command, one can select the size again by pressing **Space** afterwards.

Command	<code>\,</code>	<code>\:</code>	<code>\;</code>	<code>\quad</code>	<code>\qquad</code>	<code>\!</code>
Number of Space keystrokes after inserting the protected space	0	1	2	3	4	5
Result	AB	$A B$	$A B$	$A B$	$A B$	AB

The last size seem to produce no space. It is displayed red in LyX contrary to the other sizes, because it is a negative space. There are two more negative spaces:

Command	<code>\negmedspace</code>	<code>\negthickspace</code>
Number of Space keystrokes after inserting the protected space	6	7
Result	AB	AB

Negative spaces can lead to characters overlapping each other. Thus they can be used to enforce ligatures, what is e. g. useful for summation operators:

Command	Result
<code>\sum\sum_{f_{kl}}</code>	$\Sigma \Sigma f_{kl}$
<code>\sum\negmedspace\sum_{f_{kl}}</code>	$\Sigma \Sigma f_{kl}$

Relations like for example equal signs, are always surrounded by space. To suppress this, the equal sign is placed into a T_EX-brace. The following example demonstrates this:

$$\begin{array}{ll} \text{normal equation} & A = B \\ \text{equation without space} & A=B \end{array}$$

The command for the last formula is: `A\{=\rightarrow B`

Spaces are needed for physical units, because the space between the value and the unit is the smallest one and not a normal space. For units in text, the smallest space is inserted via the menu **Insert**▷**Formatting**▷**Thin Space** (shortcut **Ctrl+Shift+Space**).

An example to visualize the difference:

24 kW·h space between value and unit
 24kW·h smallest space between value and unit

9. Boxes and Frames

Boxes for text are described in chapter `Boxes` in the `Embedded Objects` manual.

9.1. Boxes with Frame

It is possible to frame formulas or parts of it with the commands `\fbox` and `\boxed`.

When one of these commands is inserted to a formula, a blue box appears within a frame to enter formula parts. For `\fbox` an additional formula has to be created by `Ctrl+M` within this box, because the box content will otherwise be treated as mathematical text. When `\boxed` is used, a new formula is automatically created inside the frame.

The command `\fbox` is not suitable to frame displayed formulas because the formula will always be set in the size of the text. `\boxed` is in contrary not suitable to frame inline formulas, because the formula will always be set in the size of a displayed formula.

As extension to `\fbox` there is the command `\framebox` where additionally the frame width and the alignment can be specified. `\framebox` is used in the following scheme:

`\framebox[frame width][position]{box content}`

The position can either be *l* or *r*. *l* left aligns, *r* right aligns the formula in the box. When no position is given, the formula will be centered.

Is no width given, also no position can be given. In this case the frame width is adjusted to the box content like for `\fbox`.

When the command `\framebox` is inserted, a box appears containing three blue boxes. The first two boxes are surrounded by brackets and denote the two optional arguments. The third box is for formula parts like for `\fbox`.

Command	Result
<code>\fbox_Ctrl+M \int_A=B</code>	$\int A = B$
<code>\boxed_\int A=B</code>	$\int A = B$
<code>A+\fbox_B</code>	$A + \boxed{B}$
<code>\framebox_20mm→→Ctrl+M \frac_A↓B</code>	$\frac{A}{B}$

The frame thickness can also be adjusted. To do this the following commands have to be inserted in `TEX`-mode before the formula

`\fboxrule “thickness” \fboxsep “distance”`

“distance” specifies the minimal distance between the frame and the first character in the box. An example for this is the following framed formula:

$$A + B = C$$

Before this formula the commands

`\fboxrule 2mm \fboxsep 3mm`

were inserted in $\text{T}_\text{E}\text{X}$ -mode. The given values are used for all following boxes.

To return to the standard frame size, the command

`\fboxrule 0.4pt \fboxsep 3pt`

is inserted in $\text{T}_\text{E}\text{X}$ -mode before the next formula.

9.2. Boxes without Frame

For boxes without a frame there are the following box commands: `\mbox`, `\makebox` and `\raisebox`

With `\raisebox` a box can be super- or subscripted. But in contrary to normal super- and subscripting, the characters in the box keep their font size. `\raisebox` is used in the following scheme:

`\raisebox{height}{box content}`

When the box should contain a formula, an extra formula is needed like for `\fbox`.

Note: For `\raisebox` this extra formula is created by pressing **Ctrl+M** twice instead of once because $\text{L}_\text{Y}\text{X}$ doesn't yet support `\raisebox` directly.

Command	Result
<code>H\raisebox{2mm}\{al\to lo</code>	$H^{al}lo$
<code>H\raisebox{-2mm}\{al\to lo</code>	$H_{al}lo$
<code>A=\raisebox{-2mm}\{\text{Ctrl+M Ctrl+M } \sqrt{\square} B</code>	$A = \sqrt{B}$

The command `\mbox` is equivalent to `\fbox` and `\makebox` is equivalent to `\framebox`, with the difference that there is no frame.

9.3. Colored Boxes

To be able to use all commands explained in this section, the \LaTeX -package **color**¹⁶ has to be loaded in the \LaTeX -preamble with the line¹⁷

```
\usepackage{color}
```


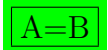
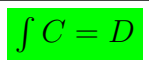
To color boxes, the command \colorbox is used in the following scheme:

```
\colorbox{color}{box content}
```

The box content can also be a box and a \colorbox can also be part of another box (see the 2nd and 3rd example). When the box should contain a formula, an extra formula has to be created, the same way as for \raisebox .¹⁸

One of the following predefined colors can be chosen:

black, blue, cyan, green, magenta, red, white and yellow

Command	Result
$\text{\colorbox}{yellow}\rightarrow\{A=B$	
$\text{\colorbox}{green}\rightarrow\{\text{\fbox}A=B$	
$\text{\fbox}\text{\colorbox}{green}\rightarrow\{\text{\Ctrl+M Ctrl+M \int}C=D$	

\colorbox only colors the box but not the characters in the box. To color all characters, the whole formula is highlighted and the wanted color is chosen in the **Text Style** dialog. The dialog can be called with the toolbar button *ab* or the menu **Edit** \triangleright **Text Style** \triangleright **Customized**. The formula number has then the same color as the formula. When the formula number should get another color than the formula characters, the color must be changed within the formula.

An example:

$$\int A = B \tag{1}$$

$$\int A = B \tag{2}$$

¹⁶The \LaTeX -package **color** is part of every \LaTeX standard installation.

¹⁷When text is colored somewhere in the document with a predefined color, \LyX loads the \LaTeX -package **color** automatically. Thus it is possible that the package is loaded twice, but this doesn't arise problems.

¹⁸This also applies for the command \fcolorbox .

Formula (1) is completely colored red.

Formula (2) was first completely colored green to set the color for the formula number. Subsequently the formula characters were colored red.

To color the frame different than the rest of the box, the command `\fcolorbox` is used in the following scheme:

`\fcolorbox{frame color}{color}{box content}`

So `\fcolorbox` is an extension of the command `\colorbox`. The frame width is set, like for `\framebox`, with `\fboxrule` and `\fboxsep`. An example:



This formula was created with the command
`\fcolorbox{cyan}{magenta}{A=B}`.

To use other colors than the predefined ones, they have to be defined first.

One can for example define the color “darkgreen” with the \LaTeX -preamble line:

`\definecolor{darkgreen}{cmyk}{0.5, 0, 1, 0.5}`

cmyk is the color space that denotes the colors **cyan**, **magenta**, **yellow** and **black**. The four comma separated numbers are the portion factor for the corresponding colors of the color space. The factors can be in the range of 0-1. Instead of **cmyk** also the color space **rgb** can be used for definitions. **rgb** denotes **red**, **green** and **blue**, so that there are in this case three portion factors for the corresponding colors. Furthermore there is the color space **gray** with one portion factor for the gray value.

As example a framed box with the new defined color **darkgreen** where the characters have been colored **yellow**:


$$\int A dx = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)} \quad (3)$$

Self-defined colors can also be used for text with the help of the command `\textcolor`:

This sentence is “darkgreen”.

`\textcolor` is used in the scheme `\textcolor{color}{characters to color}`.

9.4. Paragraph Boxes

A box that can contain several lines and paragraphs, a so called paragraph box (parbox), can be created with the menu **Insert** > **Box** or the toolbar button .

The following example shows a framed parbox in a line:

This is a line This is a paragraph box.
It is exactly 5 cm long and
can also contain formulas:
 $\int A ds = C$ with a parbox.

Such a box is created by right-clicking on the gray box inset. A dialog pops up showing the box properties. In our case set: *Decoration*: Rectangular box, *Inner Box*: Parbox, *Width*: 5 cm, *Vertical Box Alignment*: Middle

In L^AT_EX a parbox is created with the command `\parbox` that has the following scheme:

`\parbox[position]{width}{box content}`

The positions *b* and *t* are possible. *b* for bottom means that the box is aligned within the surrounding text with its last line. With *t* for top this is done with the first line. When no position is given, the box will be vertically centered, see section *Boxes* of the *Embedded Objects* manual for examples.

To frame formulas completely, including the formula number, the formula must be set into a parbox. To do this, the command `\fbox{\parbox{\linewidth-2\fboxsep-2\fboxrule}{formula}}` is inserted in T_EX-mode before the formula. `\linewidth` is hereby the line width set for the document. Because the frame is outside the parbox, 2 times the frame separation and the frame thickness must be subtracted from the line width. As this is not automatically done by L^AX due to a bug¹⁹, T_EX-mode has to be used. To be able to multiply and subtract in arguments, the L^AT_EX-package `calc`²⁰ must be loaded in the L^AT_EX-preamble with the line

`\usepackage{calc}`

Behind the formula both boxes are closed by entering `}}` in T_EX-mode. Here is an example:

$$\int A dx = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)} \quad (4)$$

As a parbox is used as argument of `\fbox`, there is in this case no difference between `\fbox` and `\boxed`.

¹⁹[LyX-bug #4483](#)

²⁰`calc` is part of every L^AT_EX standard installation.

Paragraph boxes are very useful to comment formulas directly. To do this, `\parbox` is used in combination with the command `\tag`. (more about `\tag` see sec. 19.4)

An example of a formula commented with `\parbox`:

$$5x - 7b = 3b$$

This is a description. It is distinctly separated from the formula and multiline.

Such a formula must be inserted completely in $\text{T}_{\text{E}}\text{X}$ -mode because $\text{L}_{\text{Y}}\text{X}$ does not yet support the command `\parbox` in formulas. The formula is created with the following command sequence:

The command `\[5x-7b=3b\tag*\{\parbox{5cm}\}` is inserted in $\text{T}_{\text{E}}\text{X}$ -mode.²¹ Then the description follows as normal text and finally `]\]` is inserted in $\text{T}_{\text{E}}\text{X}$ -mode. The commands `\[` and `\]` hereby create a displayed formula.

The advantages of `\parbox` can be seen in this example that was “commented” using the mathematical textmode:

$$5x - 7b = 3b \text{This is a description. It is not separated from the formula ...}$$

10. Operators

10.1. Big Operators

To be able to use all integral operators listed here, the option `Use esint package` automatically must be set in the document settings under `Math Options`.

Command	Result	Command	Result
<code>\int</code>	\int	<code>\sum</code>	Σ
<code>\oint</code>	\oint	<code>\prod</code>	Π
<code>\ointctrlockwise</code>	\oint	<code>\coprod</code>	\amalg
<code>\ointclockwise</code>	\oint	<code>\bigodot</code>	\odot
<code>\sqint</code>	\int	<code>\bigotimes</code>	\otimes
<code>\fint</code>	\int	<code>\bigoplus</code>	\oplus
<code>\landupint</code>	\int	<code>\bigwedge</code>	\wedge
<code>\landdownint</code>	\int	<code>\bigvee</code>	\vee
<code>\bigcap</code>	\cap	<code>\bigsqcup</code>	\sqcup
<code>\bigcup</code>	\cup	<code>\biguplus</code>	\uplus

All big operators can also be inserted via the math toolbar button .

²¹When the formula style `Indented` is used, `\tag*\{` can also be replaced by `\hfill`. (formula styles see sec. 17)

The operators are called big because they are bigger than the sometimes equal looking binary operators. All big operators can have limits as described in the next subsection.

For all integral operators there is a second version available, ending on **op**: `\intop`, `\ointop` etc.. These operators are different from `\int` etc. in the style the operator limits are displayed, see sec. 10.2.

Advice for Integrals

The letter *d* in an integral is an operator, that therefore has to be set upright. This is done by highlighting the *d* and using the keyboard shortcut **Alt+C R²²**. Finally the smallest space is inserted before the *d*, as this is usual for operators. An example:

incorrect: $\int A(x)dx$
 correct: $\int A(x) \, dx$

For multiple integrals there are the following commands:

Command	Result
<code>\iint</code>	\iint
<code>\oiint</code>	\oiint
<code>\sqiint</code>	\sqiint

Command	Result
<code>\iiint</code>	\iiint
<code>\iiiint</code>	\iiiint
<code>\dotsint</code>	$\int \cdots \int$

10.2. Operator Limits

Limits are created by super- and subscripts:

Command	Result
<code>\prod^{\infty}_{\square \rightarrow 0} A(x)</code>	$\prod_0^\infty A(x)$

Limits of inline formulas are set right beside the operator. Limits in displayed formulas are set above or below the operator, except for integral limits.

To force that the limits are set beside the operator, the cursor is set directly behind the operator and the limits type is changed with the menu **Edit > Math > Change Limits Type** to **Inline** (shortcut **Alt+M L**). An example:

The default limits type is this:

$$\sum_{x=0}^{\infty} \frac{1}{x^2}$$

²²Font styles see sec. 11.1

This is how it looks when the limits type was changed to **Inline**:

$$\sum_{x=0}^{\infty} \frac{1}{x^2}$$

For integrals, except those ending with **op** like `\intop`, `\ointop` etc., the limits are by default set beside the operator. But for multiple integrals the limits are often set below the operator. In the following example the limits type was therefore set to **Display** and so set below the integrals:

$$\iiint_V X \, dV = U \tag{5}$$

To specify conditions for limits, the commands `\subarray` and `\substack` are used. To create for example this expression

$$\sum_{\substack{0 < k < 1000 \\ k \in \mathbb{N}}}^n k^{-2} \tag{6}$$

the following has to be done:

First the command `\sum^{\hat{n}}_{\square}` is typed in. One is now in a blue box under the summation operator and insert there the command `\subarray_{\square}`. The blue box is now within a purple box and now several lines can be written among each other. A new line is created by inserting a line break (**Ctrl+Return**). When now

0 < k < 1000 Ctrl+Return

is typed in, a new box appears below for the new line.

The alignment of the lines can be changed to left aligned with the **table toolbar** or the menu **Edit > Rows & Columns**. To get right alignment, `\hfill_{\square}` is inserted at the beginning of the line.

The command `\substack` is equivalent to `\subarray` with the difference that the lines are always centered.

Like in formula (6) there can be too much space beside an operator, because the characters following the operator are set beside the limits.

To avoid this, the following macro can be used in the \LaTeX -preamble:

```
\def\clap#1{\hbox to 0pt{\hss #1\hss}}
\def\mathclap {\mathpalette \mathclapinternal}
\def\mathclapinternal #1#2{\clap{\$ \mathsurround =0pt #1{\#2}\$}}
```

This defines the command `\mathclap` that sets the width of the limit to 0pt. The command scheme is

```
\mathclap{limit}
```

where the limit can consist of several conditions.

Applied on formula (6), one uses the command

`\sum_{\mathclap{\substack{0<k<1000}}}` **Ctrl+Return**

to create the lower limit. The summand is now directly behind the summation operator:

$$\sum_{\substack{0 < k < 1000 \\ k \in \mathbb{N}}}^n k^{-2}$$

How to use one limit for several operators is described in sec. 10.4.

10.3. Binary Operators

Binary operators are surrounded by space when there is a character before and behind them.

Command	Result	Command	Result	Command	Result
<code>+</code>	$+$	<code>\nabla</code>	∇	<code>\oplus</code>	\oplus
<code>-</code>	$-$	<code>\bigtriangledown</code>	∇	<code>\ominus</code>	\ominus
<code>\pm</code>	\pm	<code>\bigtriangleup</code>	\triangle	<code>\otimes</code>	\otimes
<code>\mp</code>	\mp	<code>\Box</code>	\square	<code>\oslash</code>	\oslash
<code>\cdot</code>	\cdot	<code>\cap</code>	\cap	<code>\odot</code>	\odot
<code>\times</code>	\times	<code>\cup</code>	\cup	<code>\amalg</code>	\amalg
<code>\div</code>	\div	<code>\dagger</code>	\dagger	<code>\uplus</code>	\uplus
<code>*</code>	$*$	<code>\ddagger</code>	\ddagger	<code>\setminus</code>	\setminus
<code>\star</code>	\star	<code>\wr</code>	\wr	<code>\sqcap</code>	\sqcap
<code>\circ</code>	\circ	<code>\bigcirc</code>	\bigcirc	<code>\sqcup</code>	\sqcup
<code>\diamond</code>	\diamond	<code>\wedge</code>	\wedge	<code>\triangleleft</code>	\triangleleft
<code>\bullet</code>	\bullet	<code>\vee</code>	\vee	<code>\triangleright</code>	\triangleright

All binary operators can also be inserted via the math toolbar button \pm .

To typeset the Laplace operator also `\Delta` or `\nabla^2` (∇^2) can be used instead of `\bigtriangleup`.

The character Menu Separator from the menu Insert \triangleright Special Character is the operator `\triangleright`.

10.4. Self-defined Operators

With the help of the command `\DeclareMathOperator` custom operators can be defined in the \LaTeX -preamble. Its command scheme is:

`\DeclareMathOperator{new command}{display}`

Display can be characters or symbols that define how the operator looks in the output. To define a big operator a * is set behind the command. All self-defined big operators can have limits as described in sec. 10.2.

For example the L^AT_EX-preamble line

`\DeclareMathOperator*{\Lozenge}{\blacklozenge}`

defines the command `\Lozenge`, that inserts a big operator consisting of the lozenge symbol from sec. 13.2:

$$\prod_{n=1}^{\infty}$$

The command for this formula is: `\Lozenge^{\infty}\rightarrow_{n=1}`

When self-defined operators are not used several times in the document, they can also be defined with the commands `\mathop` and `\mathbin`, which have the following scheme:

`\mathop{display}` and `\mathbin{display}`

`\mathop` defines big operators, `\mathbin` binary operators.

`\mathop` can e. g. be used to use one limit for several operators:

$$\sum_{i,j=1}^N$$

The command for the formula above is:

`\mathop{\sum}\negmedspace\sum_{\rightarrow}^{\wedge N}_{i,j=1}`

11. Fonts

11.1. Font Styles

Latin letters in formulas can be set in one of the following font styles:

Command	Result	shortcut
<code>\mathbb_{ }ABC</code>	A B C	Alt+C C
<code>\mathbf_{ }AbC</code>	A b C	Ctrl+B
<code>\boldsymbol_{ }AbC</code>	<i>A</i> <i>b</i> <i>C</i>	Ctrl+Alt+B, Alt+C B
<code>\mathcal_{ }ABC</code>	<i>A</i> <i>B</i> <i>C</i>	Ctrl+E
<code>\mathfrak_{ }AbC</code>	<i>A</i> <i>b</i> <i>C</i>	-

Command	Result	shortcut
<code>\mathit</code> AbC	<i>AbC</i>	-
<code>\mathrm</code> AbC	AbC	Alt+C R
<code>\mathsf</code> AbC	AbC	Alt+C S
<code>\mathtt</code> AbC	AbC	Ctrl+Shift+P

Note: The styles `\mathbb` and `\mathcal` can only be used for big letters.

Predefined is the style `\mathnormal`.

The style commands work also for letters in mathematical constructs:

$$\mathfrak{a} = \frac{\mathfrak{b}}{\mathfrak{c}}$$

Characters in mathematical text don't appear in a math font style but in the text font style `\textrm`. That their style can't be set correctly via the text style dialog is a bug in LyX.²³

Instead of the style commands the dialog **Edit** > **Math** > **Text Style** or the toolbar button  can be used.

11.2. Bold Formulas

To make a complete formula bold, the command `\mathbf` from the previous subsection cannot be used, because it doesn't work for small Greek letters. Furthermore it prints Latin letters always upright, like in the following equation:

$$\int_n^2 \mathbf{f}(\theta) = \mathbf{\Gamma} \quad \text{equation with } \mathbf{\backslash mathbf}$$

To display the formula correctly, the command `\boldsymbol` is used:

$$\int_n^2 \boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{\Gamma} \quad \text{equation with } \mathbf{\backslash boldsymbol}$$

It is also possible to set the formula in a **boldmath environment**. This environment is created by inserting the command `\boldmath` in TeX-mode. To end the environment, the command `\unboldmath` is inserted in TeX-mode.

$$\int_n^2 \boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{\Gamma} \quad \text{equation in a boldmath environment}$$

²³LyX-bug #4629

11.3. Colored Formulas

Formulas can be colored like normal text: Highlight a formula or a formula part and use the **Text Style** dialog. Here is a formula in magenta:

$$\int A dx = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)}$$

You can also define your own colors as described in sec. 9.3. They can be used with the $\text{T}_{\text{E}}\text{X}$ code command `\textcolor` in the scheme

`\textcolor{color}{characters or formula}`

The following example was colored completely dark green and partly red:

$$\int A dx = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)}$$

Due to a bug in $\text{L}_{\text{Y}}\text{X}$ only complete formulas can be colored with self-defined colors.²⁴

11.4. Font Sizes

For characters in formulas there are, analog to characters in text, the following size commands:

`\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\normalsize`, `\small`,
`\footnotesize`, `\scriptsize` and `\tiny`

The size produced by the commands depends on the document font size, that corresponds with the command `\normalsize`. The other commands produce smaller or larger sizes than `\normalsize`. The font size can however not exceed a certain value. Is for example the document font size 12pt, the command `\Huge` switches to the same size as `\huge`.

A size command is inserted in $\text{T}_{\text{E}}\text{X}$ -mode before the formula and sets the size for all following formula and text characters. To switch back to the initial size, the command `\normalsize` is inserted behind the formula in $\text{T}_{\text{E}}\text{X}$ -mode.

²⁴[LyX-bug #5269](#)

Within a formula the size can be changed using the following size commands:

Command	Result
<code>\displaystyle</code>	$E_{\text{pot}_1} = \frac{K}{l + \frac{m}{n_2}}$
<code>\textstyle</code>	$E_{\text{pot}_1} = \frac{K}{l + \frac{m}{n_2}}$
<code>\scriptstyle</code>	$E_{\text{pot}_1} = \frac{K}{l + \frac{m}{n_2}}$
<code>\scriptscriptstyle</code>	$E_{\text{pot}_1} = \frac{K}{l + \frac{m}{n_2}}$

After entering these commands, a blue box appears in which the formula parts are inserted.

There is a further method to change the font size, that though only works for symbols or letters in mathematical text. To use it, one of the above text size commands is inserted in mathematical text. All following characters until the end of the mathematical text or until another size command will have the selected size. Two examples:

$$A = \frac{B}{c} \cdot \text{✠}$$

$$\text{✠}A\text{✠}A_{\text{✠}A}$$

Before both formulas the command `\huge` was inserted. The command for the second formula is:

```
\maltese A Alt+M M \Large \maltese \textit A → →
Alt+M M \tiny \maltese \textit A
```

If a symbol cannot be displayed in different sizes, it will always be displayed in the default size.

12. Greek Letters

All Greek letters can also be inserted via the toolbar button .

12.1. Small Letters

Command	Result
<code>\alpha</code>	α
<code>\beta</code>	β
<code>\gamma</code>	γ
<code>\delta</code>	δ
<code>\epsilon</code>	ϵ
<code>\varepsilon</code>	ε
<code>\zeta</code>	ζ
<code>\eta</code>	η
<code>\theta</code>	θ
<code>\vartheta</code>	ϑ

Command	Result
<code>\iota</code>	ι
<code>\kappa</code>	κ
<code>\varkappa</code>	\varkappa
<code>\lambda</code>	λ
<code>\mu</code>	μ
<code>\nu</code>	ν
<code>\xi</code>	ξ
<code>o</code>	o
<code>\pi</code>	π
<code>\varpi</code>	ϖ
<code>\rho</code>	ρ

Command	Result
<code>\varrho</code>	ϱ
<code>\sigma</code>	σ
<code>\varsigma</code>	ς
<code>\tau</code>	τ
<code>\upsilon</code>	υ
<code>\phi</code>	ϕ
<code>\varphi</code>	φ
<code>\chi</code>	χ
<code>\psi</code>	ψ
<code>\omega</code>	ω

How to create upright Greek letters is explained in sec. 23.9.

12.2. Big Letters

Command	Result
<code>\Gamma</code>	Γ
<code>\Delta</code>	Δ
<code>\Theta</code>	Θ
<code>\Lambda</code>	Λ
<code>\Xi</code>	Ξ
<code>\Pi</code>	Π

Command	Result
<code>\Sigma</code>	Σ
<code>\Upsilon</code>	Υ
<code>\Phi</code>	Φ
<code>\Psi</code>	Ψ
<code>\Omega</code>	Ω

That the big Greek letters appear upright is caused by a design bug when $\text{T}_{\text{E}}\text{X}$ was developed. To get correct italic big letters, begin every command with `var`. For example the command `\varGamma` produces: Γ

12.3. Bold Letters

Greek letters cannot be set with different font styles like Latin letters. They can only be made bold with the command `\boldsymbol`.

Command	Result
<code>\Upsilon\boldsymbol\Upsilon</code>	$\Upsilon\Upsilon$
<code>\theta\boldsymbol\theta</code>	$\theta\theta$

13. Symbols²⁵

Many of the symbols listed in this section can also be inserted via the toolbar buttons ∇ and \mathbb{F} .

13.1. Mathematical Symbols

Command	Result	Command	Result	Command	Result
<code>\neg</code>	\neg	<code>\forall</code>	\forall	<code>\prime</code>	$'$
<code>\Im</code>	\Im	<code>\exists</code>	\exists	<code>\backprime</code>	\backprime
<code>\Re</code>	\Re	<code>\nexists</code>	\nexists	<code>\mho</code>	\mho
<code>\aleph</code>	\aleph	<code>\emptyset</code>	\emptyset	<code>\triangle</code>	\triangle
<code>\partial</code>	∂	<code>\varnothing</code>	\varnothing	<code>\angle</code>	\angle
<code>\infty</code>	∞	<code>\dag</code>	\dagger	<code>\measuredangle</code>	\measuredangle
<code>\wp</code>	\wp	<code>\ddag</code>	\ddagger	<code>\sphericalangle</code>	\sphericalangle
<code>\imath</code>	\imath	<code>\complement</code>	\complement	<code>\top</code>	\top
<code>\jmath</code>	\jmath	<code>\Bbbk</code>	\mathbb{k}	<code>\bot</code>	\perp

13.2. Miscellaneous Symbols

Command	Result	Command	Result	Command	Result
<code>\flat</code>	\flat	<code>\hbar</code>	\hbar	<code>\diamondsuit</code>	\diamondsuit
<code>\natural</code>	\natural	<code>\hslash</code>	\hslash	<code>\Diamond</code>	\Diamond
<code>\sharp</code>	\sharp	<code>\clubsuit</code>	\clubsuit	<code>\heartsuit</code>	\heartsuit
<code>\surd</code>	\surd	<code>\spadesuit</code>	\spadesuit	<code>\P</code>	\P
<code>\checkmark</code>	\checkmark	<code>\bigstar</code>	\bigstar	<code>\copyright</code>	\copyright
<code>\yen</code>	\yen	<code>\blacklozenge</code>	\blacklozenge	<code>\circledR</code>	\circledR
<code>\pounds</code>	\pounds	<code>\blacktriangle</code>	\blacktriangle	<code>\maltese</code>	\maltese
<code>\\$</code>	$\$$	<code>\blacktriangledown</code>	\blacktriangledown	<code>\diagup</code>	\diagup
<code>\S</code>	\S	<code>\bullet</code>	\bullet	<code>\diagdown</code>	\diagdown

More symbols are listed in sec. 16.4.

Some symbols can be displayed in different sizes, see sec. 11.4.

²⁵A list with all symbols of most of the L^AT_EX-packages can be found in [4].

13.3. The Euro-Symbol €

To use the Euro symbol in formulas, the L^AT_EX-package **eurosym** must be installed and loaded with the L^AT_EX-preamble line

```
\usepackage[gennarrow]{eurosym}
```

The Euro symbol can now be inserted with the command `\euro`.

The Euro symbol can directly be inserted with the € key in mathematical text, without having **eurosym** installed. When **eurosym** is installed, `\euro` can also be inserted in T_EX-mode. The official currency symbol can then be inserted with the command `\officialeguro`, that is only available in T_EX-mode.

An overview about the different Euro symbols:

	Command	Result
formula	<code>\euro</code>	€
mathematical text	€	€
T _E X-mode	<code>\officialeguro</code>	€

14. Relations

All relations can also be inserted via the toolbar button .

Command	Result	Command	Result	Command	Result
<code><</code>	$<$	<code>=</code>	$=$	<code>></code>	$>$
<code>\le</code>	\leq	<code>\not=</code>	\neq	<code>\ge</code>	\geq
<code>\ll</code>	\ll	<code>\equiv</code>	\equiv	<code>\gg</code>	\gg
<code>\prec</code>	\prec	<code>\sim</code>	\sim	<code>\succ</code>	\succ
<code>\preceq</code>	\preceq	<code>\simeq</code>	\simeq	<code>\succeq</code>	\succeq
<code>\subset</code>	\subset	<code>\approx</code>	\approx	<code>\supset</code>	\supset
<code>\subseteq</code>	\subseteq	<code>\cong</code>	\cong	<code>\supseteq</code>	\supseteq
<code>\sqsubseteq</code>	\sqsubseteq	<code>\bowtie</code>	\bowtie	<code>\sqsupseteq</code>	\sqsupseteq
<code>\in</code>	\in	<code>\notin</code>	\notin	<code>\ni</code>	\ni
<code>\vdash</code>	\vdash	<code>\perp</code>	\perp	<code>\dashv</code>	\dashv
<code>\smile</code>	\smile	<code>\propto</code>	\propto	<code>\frown</code>	\frown
<code>\lhd</code>	\triangleleft	<code>\asymp</code>	\asymp	<code>\rhd</code>	\triangleleft
<code>\unlhd</code>	\triangleleft	<code>\doteq</code>	\doteq	<code>\unrhd</code>	\triangleleft
<code>\gtrless</code>	\gtrless	<code>\circeq</code>	\circeq	<code>\lessgtr</code>	\lessgtr
<code>\mid</code>	$ $	<code>\models</code>	\models	<code>\parallel</code>	\parallel
<code>\nmid</code>	\nmid	<code>\widehat{=}</code>	$\widehat{=}$	<code>\nparallel</code>	\nparallel

The characters `\lhd` and `\rhd` are bigger than the equal looking operators `\triangleleft` and `\triangleright`, respectively.

Relations are, in contrary to symbols, always surrounded by space.

Relations with labels can be created with the command `\stackrel`:

Command	Result
<code>A(r)\stackrel{r}{\approx} B</code>	$A(r) \stackrel{r}{\approx} B$

15. Functions


15.1. Predefined Functions

In general, variables are set *italic* in mathematical expressions, but not function names, because *sin* could be misunderstood as $s \cdot i \cdot n$. Therefore there are predefined functions, that are additionally a bit separated from prefactors. They are inserted as commands starting with a backslash before their name.

Command	Result	Command	Result
<code>Asin(x)+B</code>	$Asin(x) + B$	<code>A\sin(x)+B</code>	$A \sin(x) + B$

The following functions are predefined:

Command	Command	Command	Command
<code>\sin</code>	<code>\sinh</code>	<code>\arcsin</code>	<code>\sup</code>
<code>\cos</code>	<code>\cosh</code>	<code>\arccos</code>	<code>\inf</code>
<code>\tan</code>	<code>\tanh</code>	<code>\arctan</code>	<code>\lim</code>
<code>\cot</code>	<code>\coth</code>	<code>\arg</code>	<code>\liminf</code>
<code>\sec</code>	<code>\min</code>	<code>\deg</code>	<code>\limsup</code>
<code>\csc</code>	<code>\max</code>	<code>\det</code>	<code>\Pr</code>
<code>\ln</code>	<code>\exp</code>	<code>\dim</code>	<code>\hom</code>
<code>\lg</code>	<code>\log</code>	<code>\ker</code>	<code>\gcd</code>

They can also be inserted with the math toolbar button  `\tanexp`.

15.2. Self-defined Functions

To use a function that is not predefined, like for example the sign function $\text{sgn}(x)$, there are two possibilities:

- Define the function by inserting the following line to the L^AT_EX-preamble²⁶

`\DeclareMathOperator{\sgn}{sgn}`

Now the new defined function can be called with the command `\sgn`.

- Write the formula as usual, mark the formula name, in our example the letters *sgn*, and change it to mathematical text. At last a space is inserted between prefactor and function.

The result is the same with both methods as with a predefined function²⁷:

Command	Result
<code>A\sgn(x)+B</code>	$A \operatorname{sgn}(x) + B$
<code>A\, \underbrace{\operatorname{sgn}}_{\text{Alt+MM}}(x)+B</code>	$A \operatorname{sgn}(x) + B$

The first method is more suitable when the self-defined function should be used several times.

15.3. Limits

For limits there are defined besides `\lim`, `\liminf` and `\limsup` furthermore the following functions:

Command	Result
<code>\varliminf</code>	\varliminf
<code>\varlimsup</code>	\varlimsup
<code>\varprojlim</code>	\varprojlim
<code>\varinjlim</code>	\varinjlim

The limit is created by inserting a subscript. It is set right beside the function in an inline formula:

Command	Result
<code>\lim_x\to_A x=B</code>	$\lim_{x \rightarrow A} x = B$

In a displayed formula the limit is set below the formula, as usual:

$$\lim_{x \rightarrow A} x = B$$

²⁶For more about `\DeclareMathOperator` see sec. 10.4.

²⁷In L^AT_EX self-defined functions are displayed red, predefined ones black.

15.4. Modulo-Functions

The modulo-function is special, because it exists in four variants.

The variants in a displayed formula:

<u>Command</u>	<u>Result</u>
$a \backslash \operatorname{mod} b$	$a \operatorname{mod} b$
$a \backslash \operatorname{pmod} b$	$a \pmod{b}$
$a \backslash \operatorname{bmod} b$	$a \operatorname{mod} b$
$a \backslash \operatorname{pod} b$	$a \operatorname{ (} b)$

In an inline formula less space is set before the function names for all variants.

16. Special Characters

16.1. Special Characters in Mathematical Text

The following commands can only be used in mathematical text or in $\text{T}_{\text{E}}\text{X}$ -mode:

Command	Result	command	Result
$\backslash \operatorname{oe}$	œ	$\backslash \operatorname{o}$	ø
$\backslash \operatorname{OE}$	Œ	$\backslash \operatorname{O}$	Ø
$\backslash \operatorname{ae}$	æ	$\backslash \operatorname{l}$	ł
$\backslash \operatorname{AE}$	Æ	$\backslash \operatorname{L}$	Ł
$\backslash \operatorname{aa}$	å	$\backslash \operatorname{!}$	ı
$\backslash \operatorname{AA}$	Å	$\backslash \operatorname{?}$	İ
$\backslash \operatorname{i}$	ı	$\backslash \operatorname{j}$	Ĳ

The characters Å and Ø can also be inserted via the math toolbar button $\overset{F}{\square}$.

An exception are the commands $\backslash \operatorname{!}$ and $\backslash \operatorname{?}$, because they can be inserted in $\text{L}_{\text{Y}}\text{X}$ directly to text.

16.2. Accents in Text

With the following commands all letters can be accented. The commands must be inserted in $\text{T}_{\text{E}}\text{X}$ -mode.

Command	Result
<code>\“e</code>	ë
<code>\‘e</code>	è
<code>\^_e</code>	ê
<code>\=e</code>	ē
<code>\u_e</code>	ě
<code>\b_e</code>	ē
<code>\t_ee</code>	ēē

Command	Result
<code>\H_e</code>	é
<code>\’e</code>	é
<code>\~e</code>	ẽ
<code>\.e</code>	ė
<code>\v_e</code>	ě
<code>\d_e</code>	ę
<code>\c_e</code>	ę

With the command `\t` also two different characters can be accented. The command `\t_sz` creates: $s\hat{z}$

The accents ‘, ’ and ^ can in combination with vowels directly be inserted with the keyboard without using T_EX-mode. The same applies for the tilde²⁸ ~ in combination with *a*, *n*, or *o*.

The commands `\b`, `\c`, `\d`, `\H`, `\t`, `\u`, `\v` and accents that are inserted directly with the keyboard are also available in mathematical text. For the other accents there are special math commands to be used in formulas, see sec. 7.1.

Furthermore, with the command `\textcircled` all numbers and letters can be set into a circle, quasi accented with a circle, similar to the copyright symbol.

Command	Result
<code>\textcircled{w}</code>	Ⓜ
<code>\Large \textcircled{\normalsize\protect\raisebox{-1.5pt}{W}}</code>	Ⓜ

One has to take care that the character fits in the circle. `\Large`²⁹ specifies thereby the size of the circle. With the help of `\raisebox`³⁰ the character can be centered.

16.3. Minuscule Numbers

Minuscule numbers are created with the command `\oldstylenums`. The command can be used in formulas and in T_EX-mode. The command scheme is:

`\oldstylenums{number}`

The command `\oldstylenums{0123456789}` produces: 0123456789

²⁸This only applies for keyboards where the tilde is defined as accent.

²⁹see sec. 11.4

³⁰see sec. 9.2

16.4. Miscellaneous special Characters

The following characters can only be inserted to formulas by using commands:

Command	Result
$\widehat{\square}$	$\hat{\square}$
$\underline{\square}$	$\underline{\square}$
$\widehat{\square}\backslash\text{circ}$	$\overset{\circ}{\square}$

The degree sign $^\circ$ can nevertheless be directly inserted if the \LaTeX -preamble contains the following line³¹:

```
\DeclareInputtext{176}{\ifmmode\widehat{\circ}\else\textdegree\fi}
```

17. Formula Styles

- There are two different alignment styles:

Centered is the predefined standard

Indented for this the option **fleqn** must be inserted in the menu Document \triangleright Settings under Document Class

When **Indented** is used, the indentation can be adjusted with the length $\backslash\text{mathindent}$. Should the distance be 15 mm, the following command line is inserted in the \LaTeX -preamble

```
\setlength{\mathindent}{15mm}
```

When no length is specified, the predefined value of 30 pt will be used.

- And two different numbering styles:

Right is the predefined standard

Left for this the option **leqno** must be inserted in the menu Document \triangleright Settings under Document Class

fleqn and **leqno** can also be used together. In this case both options are inserted, separated by a comma.

The chosen styles are used for all displayed formulas of the document. When both, centered and indented formulas should be created in a document, the style **Centered** is used. The indented formulas are then set in a `flalign` environment, see sec. 18.2.3.

³¹More about this is described in sec. 23.10.



18. Multiline Formulas

18.1. General

In LyX multiline formulas are created by pressing **Ctrl+Return** inside a formula. This creates either an **eqnarray environment** that is described in sec.18.3 or, when the option **Use AMS math package** in the document settings is selected, an **align environment** that is described in sec. 18.2.1.

There are other multiline formula environments that can be created via the menu **Insert▷Math**. These environments are described in the following sections.

In all multiline formula environments a new line is created by pressing **Ctrl+Return**.

To add or delete lines, the math toolbar buttons  or , respectively, or the menu **Edit▷Rows & Columns** can be used.

18.1.1. Line Separation

There is sometimes not enough space in multiline formulas between the lines:

$$\begin{aligned} B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2x^2 + 4x_0xD &= -4x^2B^2 + 4x_0xB^2 \\ 4x^2(B^2 + x_0^2) + 4x_0x(D - B^2) + B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) &= 0 \end{aligned}$$

In L^AT_EX additional line space is specified as optional argument of the new line command. This is not yet possible in LyX³², therefore the whole formula must be inserted in T_EX-mode. To add in our example space, the command `\[3mm]` is inserted at the end of the first line. One gets:

$$\begin{aligned} B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2x^2 + 4x_0xD &= -4x^2B^2 + 4x_0xB^2 \\ 4x^2(B^2 + x_0^2) + 4x_0x(D - B^2) + B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) &= 0 \end{aligned}$$

To set the line separation for all lines in a formula, the length `\jot` is changed. The definition is: `line separation = 6 pt + \jot`. Predefined for `\jot` is the value 3 pt. To create 3 mm additional line separation as in the previous example, the command

`\setlength{\jot}{3mm+3pt}`

is inserted in T_EX-mode before the formula. This requires that the L^AT_EX-package **calc**³³ was loaded in L^AT_EX-preamble with the line

³²see [LyX-bug #1505](#)

³³**calc** is part of every L^AT_EX standard installation.

`\usepackage{calc}`

One gets:

$$\begin{aligned} B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2x^2 + 4x_0xD &= -4x^2B^2 + 4x_0xB^2 \\ 4x^2(B^2 + x_0^2) + 4x_0x(D - B^2) + B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) &= 0 \end{aligned}$$

To get back to the predefined distance, `\jot` is set to the value 3 pt.

18.1.2. Column Separation

Multiline formulas form a matrix. A formula in the `eqnarray` environment is for example a matrix with three columns. By changing the column separation in this environment, the space beside the relation sign can be changed.

The column separation is specified with the length `\arraycolsep` according to:

column separation = `2 \arraycolsep`

Thus, the command

`\setlength{\arraycolsep}{1cm}`

inserted in `TEX`-mode, sets for all following formulas a column separation of 2 cm. To get back to the predefined distance, `\arraycolsep` is set to 5 pt.

A formula with 2 cm column separation:

$$\begin{array}{ccc} A & = & B \\ C & \neq & A \end{array}$$

A formula with the predefined column separation for matrices of 10 pt:

$$\begin{array}{ccc} A & = & B \\ C & \neq & A \end{array}$$

18.1.3. Long Formulas

Long formulas can be typeset using these methods:

- When one side of the equation is much shorter than the line width, this one is chosen for the left side and the right side is typeset over two lines:

$$\begin{aligned} H &= W_{SB} + W_{mv} + W_D - \frac{\hbar^2}{2m_0}\Delta - \frac{\hbar^2}{2m_1}\Delta_1 - \frac{\hbar^2}{2m_2}\Delta_2 - \frac{e^2}{4\pi\varepsilon_0|\mathbf{r} - \mathbf{R}_1|} \\ &\quad - \frac{e^2}{4\pi\varepsilon_0|\mathbf{r} - \mathbf{R}_2|} + \frac{e^2}{4\pi\varepsilon_0|\mathbf{R}_1 - \mathbf{R}_2|} \end{aligned} \quad (7)$$

The minus sign at the beginning of the second line does normally not appear as operator because it is the first character of the line. Thus it would not be surrounded by space and could not be distinguished from the fraction bar. To avoid this, 3pt space was inserted behind the minus sign with the command `\hspace`.³⁴

- When both sides of the equation are too long, the command `\lefteqn` is used. It is inserted to the first column of the first line and effects that all further insertions overwrite the following columns:

$$\begin{aligned}
 & 4x^2 (B^2 + x_0^2) + 4x_0x (D - B^2) + B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + D^2 \\
 & \quad - B^2 - 2B\sqrt{r_g^2 - x^2 + 2x_0x - x_0^2 + r_g^2 - x^2 + 2x_0x - x_0^2} \\
 & = B^2 + 2 (r_g^2 + 2x_0x - x_0^2 - r_k^2) + \frac{(r_g^2 + 2x_0x - x_0^2 - r_k^2)^2}{B^2} \quad (8)
 \end{aligned}$$

After the insertion of `\lefteqn`, the cursor is in a purple box that is a bit shifted to the left from the blue one. In this the formula is inserted.

The content of the further lines is inserted to the second or another formula column. The greater the column number where it was inserted, the larger the indentation.

Note the following when using `\lefteqn`:

- * The formula doesn't use the full page width. When e.g. the term $-B^2$ is added to the first line in the above example, it would have been outside the page margin. To better use the width, negative space can be inserted at the beginning of the first line.
 - * Due to a bug in LyX the cursor cannot be set with the mouse into the first line.³⁵ One can only set the cursor at the beginning of the line and move it with the arrow keys.
- Other methods to set long formulas are offered by the environments described in sec. 18.5 and sec. 18.6.

18.1.4. Multiline Brackets

For brackets spanning multiple lines the following problem occurs:

$$A = \sin(x) \left[\prod_{R=1}^{\infty} \frac{1}{R} + \dots \right. \\
 \left. \dots + B - D \right]$$

³⁴more about `\hspace` see sec. 8.2

³⁵LyX-bug #1429

The closing bracket is smaller than the opening bracket because brackets with variable size may not span multiple lines.

To set the bracket size for the second line correctly, the first line is ended with `\right.` and the second line with `\left.`³⁶. After `\left.` the command `\vphantom{\prod_{R=1}^{\infty} R}` is inserted, because the multiplication operator with its limits is the largest symbol in the first line and this should be the size for the bracket in the second line.

The result is this:



$$A = \sin(x) \left[\prod_{R=1}^{\infty} \frac{1}{R} + \dots \right. \\ \left. \dots + B - D \right]$$

18.2. Align Environments

Align environments can be used for every kind of multiline formulas. They are specially useful to set several formulas side by side.

Align environments consist of columns. The odd columns are right aligned, the even ones left aligned. Every line in an Align environment can be numbered.

Align environments are created via the menu **Insert**▷**Math**. With the menu **Edit**▷**Math**▷**Change Formula Type** already existing formulas can be converted to Align environments.

To add or delete columns, the math toolbar buttons  or , respectively, or the menu **Edit**▷**Rows & Columns** can be used.

18.2.1. Standard align Environment

This Align environment is created by pressing **Ctrl+Return** in a formula or by the menu **Insert**▷**Math**▷**AMS align Environment**.

An example for two formulas set side by side, that are created with a four column align environment:

$$\begin{array}{r} A = \sin(B) \\ C \neq A \end{array} \qquad \begin{array}{r} C = D \\ B \neq D \end{array}$$

As it can be seen, the formulas in this environment are placed so as if there would be a `\hfill`³⁷ before the first and after every even column. When the formula style **Indented**³⁸ is used, the formula is set without the `\hfill` before the first column.

³⁶for more about `\left` and `\right` see sec. 5.1.2

³⁷more about `\hfill` see sec. 8.2

³⁸formula styles see sec. 17

18.2.2. Alignat Environment

The alignat environment has no predefined column separation. It can be inserted manually with the spaces that are described sec. 8.

The above example in the alignat environment where 1 cm space was inserted at the beginning of the second formula:

$$\begin{array}{rcl} A = \sin(B) & C = D \\ C \neq A & B \neq D \end{array}$$

Because the column separation can be set separately for every column, this environment is especially suitable to set three and more formulas side by side.

18.2.3. Flalign Environment

In this environment the first two columns are always set as much as possible to the left and the last two ones to the right. An example:

$$\begin{array}{rcl} A = 1 & B = 2 & C = 3 \\ X = -1 & Y = -2 & Z = 4 \end{array}$$

By creating a flalign environment with an odd number of columns where an empty $\text{T}_{\text{E}}\text{X}$ -brace is inserted to the last column, several formulas in a document can be set to the left, although the formula style **Centered** is used. As example the indented formula (5):

$$\iiint_V X \, dV = U \tag{9}$$

The first two columns contain the formula. To indent it as with the formula style **Indented**, 30 pt space was inserted at the beginning of the first column.

18.3. Eqnarray Environment

When this environment has been created, three blue boxes appear. The content of the first box is right aligned, the content of the last one left aligned. The content of the middle box appears centered and a bit smaller, because it is designed to insert there only relation characters.

$$\begin{array}{rcl} \frac{ABC}{D} & \frac{ABC}{D} & \frac{ABC}{D} \\ AB & AB & AB \\ A & = & A \end{array}$$

18.4. Gather Environment

This environment consists of only one centered column. Every line can be numbered.

$$A = 1 \tag{10}$$

$$X = -1 \tag{11}$$

18.5. Multline Environment

The multiline environment consists, like the gather environment, of only one column. But the first line is left aligned, the last one right aligned. All other lines are centered. Therefore this environment is suitable for long formulas. As example formula (8) in the multiline environment:

$$\begin{aligned} &4x^2 (B^2 + x_0^2) + 4x_0x (D - B^2) + B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + D^2 \\ &\quad - B^2 - 2B\sqrt{r_g^2 - x^2 + 2x_0x - x_0^2} + r_g^2 - x^2 + 2x_0x - x_0^2 \\ &= B^2 + 2 (r_g^2 + 2x_0x - x_0^2 - r_k^2) + \frac{(r_g^2 + 2x_0x - x_0^2 - r_k^2)^2}{B^2} \end{aligned} \tag{12}$$

In the output only the last (first) line of a multiline environment appears numbered when the document numbering is right (left).³⁹

With the commands `\shoveright` and `\shoveleft` a centered line can be right or left aligned, respectively. The commands are used as follows:

`\shoveright{line content}` and `\shoveleft{line content}`

The length `\multlinegap` specifies the distance of the first line from the left page margin. Predefined is the length 0 pt.

As example the above formula where the command

`\setlength{\multlinegap}{2cm}`

was inserted in T_EX-mode before:

$$\begin{aligned} &4x^2 (B^2 + x_0^2) + 4x_0x (D - B^2) + B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + D^2 \\ &\quad - B^2 - 2B\sqrt{r_g^2 - x^2 + 2x_0x - x_0^2} + r_g^2 - x^2 + 2x_0x - x_0^2 \\ &= B^2 + 2 (r_g^2 + 2x_0x - x_0^2 - r_k^2) + \frac{(r_g^2 + 2x_0x - x_0^2 - r_k^2)^2}{B^2} \end{aligned} \tag{13}$$

The second line was left aligned using `\shoveleft`.

³⁹numbering styles see sec. 17

18.6. Multiline Formula Parts

To display only parts of a formula with multiple lines, one of the following environments are used: **aligned**, **alignedat**, **gathered** or **split**. They can be inserted via the menu **Insert**▷**Math** or by using the commands described in this section.

The first three have the same properties as the corresponding multiline formula environments, but it is possible to set further formula parts beside them. An example:

$$\left. \begin{array}{l} \Delta x \Delta p \geq \frac{\hbar}{2} \\ \Delta E \Delta t \geq \frac{\hbar}{2} \end{array} \right\} \text{Uncertainty relations}$$

To get this formula, a displayed formula is created where the command `\aligned` is inserted. A purple box appears around the blue formula box where now columns and lines can be added. Outside the multiline environment other formula parts can be set, like the brace.

The aligned environment is also suitable for long formulas whose lines are horizontally aligned. Using `aligned` in a displayed formula has the advantage that the formula number is vertically centered behind the lines. As example formula (7) in the aligned environment:

$$\begin{aligned} H = W_{SB} + W_{mv} + W_D - \frac{\hbar^2}{2m_0}\Delta - \frac{\hbar^2}{2m_1}\Delta_1 - \frac{\hbar^2}{2m_2}\Delta_2 - \frac{e^2}{4\pi\epsilon_0|\mathbf{r} - \mathbf{R}_1|} \\ - \frac{e^2}{4\pi\epsilon_0|\mathbf{r} - \mathbf{R}_2|} + \frac{e^2}{4\pi\epsilon_0|\mathbf{R}_1 - \mathbf{R}_2|} \end{aligned} \quad (14)$$

To use the environments **alignedat**, **gathered**, or **split**, the command `\alignedat`, `\gathered`, or `\split` are inserted, respectively. The `split` environment has the same properties as the aligned environment but it can only have two columns.

18.7. Text in multiline Formulas

In the `Align` environments, and the `multiline` and `gather` environment, text can be inserted that will appear in a separate line and doesn't affect the column alignment. To do this, the command `\intertext` is used in the following scheme:

`\intertext{text}`

The text should not be longer than a line because it cannot be hyphenated. As `LyX` doesn't yet support `\intertext` directly, the text is written as mathematical text. `\intertext` must hereby be at the beginning of a line and appears in the output

above this line. An example where the text was inserted at the beginning of the second line:

$$I = a\sqrt{2} \int_0^{2\pi} \sqrt{1 + \cos(\phi)} \, d\phi \quad (15)$$

integrand is symmetric to $\phi = \pi$, therefore

$$= 2a\sqrt{2} \int_0^{\pi} \sqrt{1 + \cos(\phi)} \, d\phi \quad (16)$$

19. Formula Numbering


19.1. General

Numbered formulas can be created with the menu **Insert**▷**Math**▷**Numbered Formula** (shortcut **Ctrl+Alt N**). Existing formulas can be numbered with the menu **Edit**▷**Math**▷**Toggle Numbering** (shortcut **Alt+M N**). The formula number is displayed in L^yX behind the formula as number sign in parentheses. The number sign is replaced in the output by the formula number.


When numbering is turned on in multiline formulas, all lines will be numbered. But the numbering can be controlled with the menu **Edit**▷**Math**▷**Toggle Numbering of Line** (shortcut **Alt+M Shift+N**) for every line.

Except for inline formulas, all formulas can be numbered with two different styles, see sec. 17.

19.2. Cross-References

All labeled formulas can be cross-referenced. A label is added by the menu **Insert**▷**Label** or the toolbar button . The cursor must hereby be inside a displayed formula. A dialog pops up displaying the prefix **eq:** in a text field. The label is inserted there behind the prefix. The predefined prefix means “equation” and makes it easier to find labels in large documents because it marks it as formula label to divide it from e. g. section labels. To change a label, the menu **Insert**▷**Label** is used again.

The name of the label is displayed in L^yX within two parentheses behind formula. A formula with a label is always numbered.

Cross-references are inserted via the menu **Insert**▷**Cross-Reference** or with the toolbar button . A formula cross-reference appears in the output as formula number. When in the cross-reference dialog window the format (**<reference>**) is chosen, the cross-reference appears in the output as formula number in parentheses.

By right-clicking on a cross-reference in L^AT_EX, one jumps to the formula that is referenced.

Here are as examples cross-references to formulas of the following subsections:

The equations (something) and (17b) are equivalent. In (W) big Latin letters are used for the numbering in contrary to (XXI).

When the argument of `\tag`⁴⁰ contains a box like in sec. 9.4, the formula cannot be referenced.

19.3. Subnumbering

With the help of the commands `\begin{subequations}` and `\end{subequations}` formulas can be subnumbered. Both commands are inserted in T_EX-mode.

An example:

$$A = C - B \tag{17}$$

$$B = C - A \tag{17a}$$

$$C = A + B \tag{17b}$$

To create the example, the following is done:

1. first formula is inserted
2. `\addtocounter{equation}{-1}` `\begin{subequations}` is inserted after the first formula
3. second formula is inserted
4. third formula is inserted
5. `\end{subequations}` is inserted after the third formula

Every formula between the commands `\begin` and `\end` is subnumbered as a, b, c, ... For multiline formulas every line will be subnumbered. All subnumbered formulas are treated as *one* numbered formula. But as every numbered formula increases the counter `equation` by one, the command `\addtocounter` is needed to decrease it. Otherwise the formulas (17), (17a), (17b) would be numbered as (17), (18a), (18b).

By inserting the commands in T_EX-mode, a space is created between the first two formulas. To revert this -5 mm vertical space is inserted after the command `\begin{subequations}`. When the formula style `Indented`⁴¹ is used, -7 mm space is inserted instead.

⁴⁰`\tag` is described in sec. 19.4.

⁴¹formula styles see sec. 17

Here is an example for a multiline formula where the numbering was turned off for the second line:

$$A = (B - Z)^2 = (B - Z)(B - Z) \tag{18a}$$

$$= B^2 - ZB - BZ + Z^2$$

$$= B^2 - 2BZ + Z^2 \tag{18b}$$

19.4. User-defined Numbering

With the standard numbering parentheses are set around the formula number. To replace the parentheses for example by vertical bars, the following line is added to the L^AT_EX-preamble:

```
\def\tagform@#1{\maketag@@@{|#1|}}
```

To use other characters, the vertical bars besides the `#1` are replaced by one or more characters. To get only the formula number the vertical bars are omitted.

When there should be an expression of your choice instead of the consecutive formula number in parentheses behind the formula, the command `\tag` is used:

$$A + B = C \tag{something}$$

In this example the command `\tag_something` was inserted to the formula.

When the command `\tag*_something` is inserted instead, the star prevents the parentheses around the expression:

$$A + B = C \tag{something}$$

To restart the formula numbering with new document parts or sections, the following command is used:

```
\@addtoreset{equation}{part}
```

resp.

```
\@addtoreset{equation}{section}
```

To be able to use these commands in T_EX-mode, the “@” character has to be made “active” for L^AT_EX using the command `\makeatletter`. The command `\makeatother` reverts this. So the command sequence in T_EX-mode is:

```
\makeatletter
```

```
\@addtoreset{equation}{section}
```

```
\makeatother
```

In the L^AT_EX-preamble `\makeatletter` and `\makeatother` can be omitted as they are automatically internally inserted by L^AT_EX.

To revert `\@addtoreset`, the file `remreset.sty`⁴² has to be loaded in the L^AT_EX-preamble with the line

```
\usepackage{remreset}
```

Then the command `\@removefromreset` can be used with the same scheme as `\@addtoreset`.

Sometimes formulas should be numbered in the following form:

```
(section number.formula number)
```

The formula number should start with every section with “1”.

For this case there is the command `\numberwithin`, which is used with the following scheme:

```
\numberwithin{counter}{sectioning}
```

Counter denotes what kind of numbering is affected, sectioning denotes what number is before the dot.

Thus in our case the following L^AT_EX-preamble or T_EX-Code line is used:

```
\numberwithin{equation}{section}
```

This is the result:

$$A + B = C \tag{19.19}$$

To number e. g. tables so that the number of the part is the sectioning,

```
\numberwithin{table}{part}
```

 is used.

To go back to the standard numbering or to prevent this kind of numbering when it is defined by the document class, the following command is inserted as T_EX-Code or to the L^AT_EX-preamble:

```
\renewcommand{\theequation}{\arabic{equation}}
```

or

```
\renewcommand{\thetable}{\arabic{table}}
```

`\numberwithin` uses internally the command `\@addtoreset`, described above, that also needs to be reverted.

19.5. Numbering with Roman Numbers and Letters

Formulas can also be numbered with Roman numbers and Latin letters. To number for example with small Roman numbers, the command

```
\renewcommand{\theequation}{\roman{equation}}
```

⁴²`remreset` is part of the L^AT_EX-package `carlisle` that is part of every L^AT_EX standard installation.

is inserted before the formula in T_EX-mode. `\renewcommand` redefines the predefined command `\theequation` to the command `\roman{equation}`.⁴³ `equation` is the formula counter. When the command `\the` is used as prefix for a counter, the value of the counter is output as Arabic number. When a formula is numbered, L^AT_EX sets internally the command `\theequation` behind the formula. `\roman{equation}` outputs the counter as small Roman number.

All formulas behind the command `\renewcommand` are now numbered Roman. To switch to numbering with big Roman numbers, the command is inserted again, but `\roman` is replaced by `\Roman`. To “number” with small Latin letters there is the command `\alph`, for big ones there is the command `\Alph`.

Note: Only maximal 26 formulas can be numbered with Latin letters in one document.

A = small roman	(xx)
B = big Roman	(XXI)
C = small Latin	(v)
D = big Latin	(W)

To switch back to the default numbering, insert the command:

```
\renewcommand{\theequation}{\arabic{equation}}
```

$$E = \text{Arabic} \tag{24}$$

As you see, formulas are numbered serially independent from the numbering style. When then numbering should start with “1” when the style is changed, new equation counters have to be defined. A description about this can be found in the file [Formula-numbering.lyx](#).

20. Chemical Symbols and Equations

An example text from chemistry:

The SO_4^{2-} -ion reacts with two Na^+ -ions to sodium sulfate (Na_2SO_4). The chemical equation for this is:



⁴³The command `\renewcommand` has the same scheme like the command `\newcommand` that is described in sec. 22.1.

This chemical equation can directly be created as formula. To avoid that the symbols appear italic, everything is highlighted and changed by the shortcut `Alt+C R` to the upright font style.⁴⁴

A more convenient way to typeset chemical formulas is to use the command `\ce` that is available when the L^AT_EX-package **mhchem** is installed. After inserting `\ce` to a formula a new blue box appears where chemical formulas can be inserted in an intuitive way.

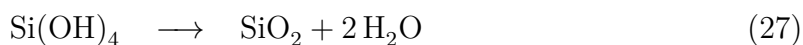
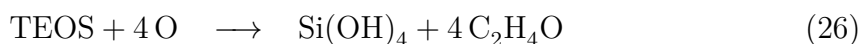
Command	Result
<code>\ce{H2CO3}</code>	H_2CO_3
<code>\ce{SO4^2-}</code>	SO_4^{2-}
<code>\ce{(NH4)2S}</code>	$(\text{NH}_4)_2\text{S}$
<code>\ce{KCr(SO4)2.12H2O}</code>	$\text{KCr}(\text{SO}_4)_2 \cdot 12 \text{H}_2\text{O}$
<code>\ce{A-B\!-\!C\!-\!D}</code>	$\text{A}-\text{B}=\text{C}\equiv\text{D}$
<code>\ce{^227_90Th+}</code>	$^{227}_{90}\text{Th}^+$
<code>\ce{CO2 + C <=> 2CO}</code>	$\text{CO}_2 + \text{C} \rightleftharpoons 2 \text{CO}$
<code>\ce{CO2 + C ->[\alpha][\beta] 2CO}</code>	$\text{CO}_2 + \text{C} \xrightarrow[\beta]{\alpha} 2 \text{CO}$

Note: Inserting a formula to a `\ce` box will lead to L^AT_EX errors. In this case T_EX code has to be used like for `\ce{\mu\text{-Cl}}`: $\mu\text{-Cl}$

Using `\ce` the command for equation (25) is:

`\ce{2Na + SO4^2- -> Na2SO4}`

To create multiline chemical equations first a multiline formula is created as described in sec. 18. Afterwards the command `\ce` is used in every small blue box of the formula. (26) and (27) are an example of a multi-stage chemical reaction where every equation has its own number.



Besides `\ce` the **mhchem** package provides the command `\cf` that has to be used for special cases. For more information about `\cf` and more examples have a look at the documentation of **mhchem**, [6].

⁴⁴font styles see sec. 11.1

21. Diagrams

LyX supports two types of commutative diagrams: **amscd** and **xymatrix** that are explained in the following.

21.1. Amscd Diagrams

Diagrams of this type visualize relations by vertical and horizontal lines or arrows:

$$\begin{array}{ccccc}
 A & \longrightarrow & B & \longrightarrow & C \\
 \uparrow & & & & \downarrow \\
 F & \longleftarrow & E & \longleftarrow & D
 \end{array}$$

To get them, the command `\CD` is inserted to a formula. A blue box appears with two dashed lines where further commands can be inserted. With **Ctrl+Return** a new line is created. Horizontal relations are inserted in odd, vertical in even formula lines.

To create the relations there are the following commands:

- `@<<<` creates a left arrow, `@>>>` a right arrow and `@=` a long equal sign
- `@AAA` creates an up arrow, `@VVV` a down arrow and `@|` a vertical equal sign
- `@.` is a placeholder for non-existent relations

All arrows can be labeled as follows:

- Is text inserted between the first and second `<` or `>`, resp., it is placed above the arrow. When it is inserted between the second and third one, it appears under the arrow.
- When text for vertical arrows is inserted between the first and second `A` or `V`, resp., it is placed left beside the arrow. When it is inserted between the second and third one, it appears right beside the arrow. If the text contains an `A` or `V`, these letters must be set into a `TeX`-brace.

As example a diagram with all possible relations:

$$\begin{array}{ccccccc}
 A & \xrightarrow{j} & B & \xrightarrow{k} & C & \longequal{\quad} & F \\
 \uparrow m & & & & \downarrow V & & \parallel \\
 D & \xleftarrow{j} & E & \xrightarrow{k} & F & \longequal{\quad} & C
 \end{array}$$

The command for this is:

```

\CD_ A@>j>>B@>k>>C@=F Ctrl+Return
 @AmAA@.@VV\{V→V@| Ctrl+Return
 D@<<j<E@>k>>F@=C
  
```

21.2. Xymatrix Diagrams

To be able to use xymatrices, the \LaTeX -package **xypic** must be installed. A xymatrix is created by inserting the command `\xymatrix` in a formula. Then you are able to add new matrix columns and rows like for normal matrices, see sec. 4.

In contrary to amscd diagrams, xymatrices supports diagonal and curved arrows, and much more. All possibilities to create commutative diagrams and decorations are explained in detail in the *XY-pic manual* that you find in the menu **Help**▷**Specific Manuals**▷**XY-pic Manual**.

22. User-defined Commands

Note: The names of user-defined commands and macros may only consist of Latin letters.

22.1. The Command `\newcommand`

Many \LaTeX -commands are too long to be used frequently. But it is possible to define with the command `\newcommand` new shorter commands.

The command scheme of `\newcommand` is:

```
\newcommand{new command name}[number of arguments][optional value]
{command definition}
```

Note: Assure that the name of the new command is not already used in your document or by \LaTeX -packages that you use. When you for example define the command `\le` for `\Leftarrow`, you get an error message because `\le` is already defined as command for “ \leq ”.

The number of arguments is an integer in the range 0-9 and specifies how many arguments the new command should have. With the optional value a value for an optional argument can be predefined. When this is done, the *first* argument of the new command is automatically an optional one.

Here are some examples:

- To define the command `\gr` for `\Longrightarrow`, the \LaTeX -preamble line is:
`\newcommand{\gr}{\Longrightarrow}`
- To define the command `\us` for `\underline`, the argument (that should be underlined) must be taken into account. For this the preamble line is:

```
\newcommand{\us}[1]{\underline{#1}}
```

The character `#` acts as argument placeholder, the `1` behind it denotes that it is the placeholder for the first argument.

- For `\framebox` one can e. g. define the command `\fb`:

```
\newcommand{\fb}[3]{\framebox#1#2{ $#3$}}
```

The two Dollar signs creates the extra formula needed for `\framebox`, see sec. 9.1.

- To create a new command for `\fcolorbox` where the color for the box needn't to be specified, the argument for the color is defined optional:

```
\newcommand{\cb}[3][white]{\fcolorbox{#2}{#1}{ $#3$}}
```

When the color is not specified when using `\cb`, the predefined color `white` will be used.

A test of the new defined commands:

Command	Result
<code>A\gr_B</code>	$A \implies B$
<code>\us{ABcd}</code>	\underline{ABcd}
<code>\fb{[2cm]}\to\{\to\{\int_A=B</code>	$\int A = B$
<code>\cb{red}\to\{\int_A=B</code>	$\int A = B$
<code>\cb{green}\{\red\to\{\int_A=B</code>	$\int A = B$

22.2. Math Macros

User-defined commands are especially convenient for complex expressions. When you are for example dealing in a document with quadratic equations, the same solution type occurs several times. The general form of a quadratic equation is

$$0 = \lambda^2 + p\lambda + q$$

The general form of the solution is


$$\lambda_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

To define a command for the solution formula where only the three parameters λ , p and q need to be specified and the index of λ can be given optionally, the L^AT_EX-preamble line is




```
\newcommand{\qG}[4][1,\,2]{#2_{#1}=-\frac{#3}{#2}\pm
\sqrt{\frac{#3^2}{#4}-#4}}
```

To create with this the solution formula, the command `\qG{\lambda}\to\{p}\to\{q}` is inserted to a formula.

The definition of the new command is unintuitive because one has to know the schemes of all used \LaTeX commands, e.g. that a fraction is inserted in \LaTeX as $\text{\frac{numerator}{denominator}}$. Furthermore one can easily forget a brace in the definition and cannot see in LyX what the new command is doing. To avoid these problems LyX offers the possibility to use math macros instead of the command \newcommand .

A math macro is created by using the menu **Insert** \triangleright **Math** \triangleright **Macro** or the toolbar button . The math macro toolbar appears together with the following box where the macro is defined:



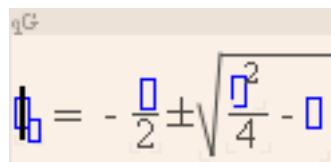
\newmacroName is the default name of the macro that should be changed to something sensible. The wanted formula is inserted in the first blue box. An argument placeholder is inserted with the command \#argumentnumber , e.g. \#1 or by using the macro toolbar button . Argument placeholders are displayed red. Maximum 9 arguments are possible. Optional arguments are created with the toolbar button . The first non-optional argument can be transformed to an optional one with the toolbar button . In the second blue box the appearance of the macro in LyX can be defined. Normally you want to see it as it is defined, so the box is kept empty. But when you have created a macro that needs lot of space on the screen, you can insert in the box for example

qG: \#1 , \#2 , \#3 , \#4

For the macro only the arguments with the macro name in front of them will then be displayed in LyX , leading to a better overview. The formula appears in the output as defined in the first box.

The appearance of macros in formulas can furthermore be changed for single macros by setting the cursor in the macro and using the menu **View** \triangleright **(Un)fold Math Macro**.

To use a macro, the macro name is inserted as command to a formula, in our case \qG . Our macro looks in LyX like this:



Here is our macro example with the arguments x , $\ln(x)$ and B :

$$x_{1,2} = -\frac{\ln(x)}{2} \pm \sqrt{\frac{\ln(x)^2}{4} - B}$$

LyX offers in the menu **Tools** \triangleright **Preferences** \triangleright **Editing** \triangleright **Control** different styles to edit macros. To find the style that suits you the most, choose a style and set the cursor in a macro formula to see the difference.

A math macro is transformed internally to a `\newcommand` command when exporting the document. The created `\newcommand` command is not placed in the \LaTeX -preamble, therefore macros can only be used in formulas that are in the document below the macro definition box.

Math macros can also be directly be created from a `\newcommand` command. When writing for example the command

```
\newcommand{\larrow}[2]{\xleftarrow[#2]{#1}}
```









in \LaTeX as normal text, highlighting it completely and using then the shortcut **Ctrl+M**, the command will be transformed to a math macro. Using this method you need to be careful that the `\newcommand` command is typed correctly, otherwise you get a faulty macro leading to \LaTeX errors.


Math macros currently yet have the problem that further formulas in macro definitions are handled wrongly. Therefore the example `\fb` from sec.22.1 cannot be created as macro.

When the cursor is in a macro definition box, you will see the macro toolbar in \LaTeX :



The macro toolbar contains from left to right the following buttons:

-  Edit \triangleright Math \triangleright Macro Definition \triangleright Remove Last Argument
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Append Argument
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Make First Non-Optional into Optional Argument
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Make Last Optional into Non-Optional Argument
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Remove Optional Argument
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Insert Optional Argument
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Remove Last Argument Spitting Out To The Right
-  Edit \triangleright Math \triangleright Macro Definition \triangleright Append Argument Eating From The Right

 Edit > Math > Macro Definition > Append Optional Argument
Eating From The Right

23. Tips

23.1. Negative Numbers

Negative numbers often look ugly in formulas because the minus sign before the number is set with the same length as the minus operator sign. When writing the negative number in normal text, the minus sign appears correctly.

Thus, the problem disappears when converting the minus sign to mathematical text.

An example to visualize the problem:

normal text: $x = -2$
formula: $x = -2$
solution: $x = -2$

23.2. Comma as decimal Separator

In \LaTeX a comma inside a formula is used, according to the English convention, as number group separator. So there will be space added behind all commas in formulas.

To avoid this, the comma is highlighted and changed to mathematical text (shortcut Ctrl+M).

To use all formula commas in the document as decimal separator, the file **icomma.sty**⁴⁵ is loaded with the \LaTeX -preamble line

```
\usepackage{icomma}
```

23.3. Physical Vectors

Predefined vectors are offered by the \LaTeX -package **braket** that is loaded with the \LaTeX -preamble line

```
\usepackage{braket}
```

The following commands are defined:

Command	Result
$\text{\textbackslash Bra}\{\psi$	$\langle\psi $
$\text{\textbackslash Ket}\{\psi$	$ \psi\rangle$
$\text{\textbackslash Braket}\{\psi \phi$	$\langle\psi \phi\rangle$

⁴⁵**icomma** is part of the \LaTeX -package **was**.

The command `\Braket` assures that all vertical bars are set in the size of the surrounding brackets:

$$\left\langle \phi \left| J = \frac{3}{2}, M_J \right. \right\rangle$$

The effect of `\Braket` can also be achieved using the command `\middle`, that is described in sec. 5.1.2.

23.4. Self-defined Fractions

To define custom commands for fractions, the command `\genfrac` is used in the following scheme:

```
\genfrac{left bracket}{right bracket}{fraction bar thickness}{style}
      {numerator}{denominator}
```

The style is a number in the range of 0-3.

Number	Style (Size)
0	display style formula
1	inline formula
2	small
3	tiny

When no style is given, the size is adjusted to the surrounding environment like for the command `\frac`.

When no fraction bar thickness is given, the predefined value of 0.4 pt will be used.

For example, the commands `\dffrac` and `\tbinom` from sec. 3.2 are defined with the commands

```
\newcommand{\dffrac}[2]{\genfrac{}{}{0}{#1}{#2}}
```



and

```
\newcommand{\tbinom}[2]{\genfrac{({})}{0pt}{1}{#1}{#2}}
```

To define a fraction where the fraction bar thickness can be given as optional argument, the following line is inserted to the L^AT_EX-preamble:

```
\newcommand{\fracS}[3][\genfrac{}{}{#1}{#2}{#3}]
```

A test:

Command	$\frac{A}{B}$	$\frac{A}{B}$
Result		

As one can see, the distance of the numerator and the denominator to the fraction bar is round about three times the bar thickness.

23.5. Canceled Formulas

To cancel formulas or formula parts, the L^AT_EX-package **cancel** has to be loaded with the L^AT_EX-preamble line

```
\usepackage[samesize]{cancel}
```

There are four ways to cancel formulas:

Command	Result
$\cancel{\int A=B}$	$\int A=B$
$\bcancel{\int A=B}$	$\int A=B$
$\xcancel{\int A=B}$	$\int A=B$
$\cancelto{1\rightarrow}{\int A=B}$	$\int A=B^1$

\cancelto is especially suitable to visualize the reduction of fractions within formulas:

$$\frac{(x_0 + bB)^2}{(1 + b^2)^2} = \frac{x_0^2 + B^2 - r_g^2}{1 + b^2}$$

23.6. Formulas in Section Headings

When formulas are used in section headings, the following has to be taken into account:

When **hyperref** support is enabled in the document settings dialog under **PDF Properties**, PDF-bookmarks are created for every section heading in the table of contents.

If a section heading contains formulas, they are incorrectly displayed in the bookmark text, because formulas in bookmarks infringe the PDF conventions.

Both problems can be solved by inserting at the end of the section heading a short title with the menu **Insert**▷**Short Title**. Short titles are used as alternative for multiline section headings to keep the table of contents clearly arranged. Only the short title appears in the table of contents and therefore also in the PDF-bookmark.

When formulas should be used in the table of contents but **hyperref** is used, one can use the following command in T_EX-mode:

```
\texorpdfstring{part}{alternative}
```

Part is the part of the heading that shouldn't appear in the PDF-bookmark. This can be characters, formulas, footnotes, but also cross-references. The alternative is used instead of the part for the bookmark.

Here are two example headings:

23.6.1. Heading without formula in table of contents $\sqrt{-1} = i$

23.6.2. Heading with formula in table of contents $\sqrt{-1} = i$

In the first heading a short title was used, in the second one `\texorpdfstring`.

To get the same formatting as for the other headings, the complete heading was set into a **boldmath** environment⁴⁶.

23.7. Formulas in multi-column Text

Formulas in multi-column text are often too wide to fit into a column and thus need to be set over the whole page width. This is done by using the L^AT_EX-package **multicol**⁴⁷, that is loaded with the L^AT_EX-preamble line

```
\usepackage{multicol}
```

Note herby that the setting **Two-column document** in the menu **Document**▷**Settings** under **Text Layout** must *not* be selected.

Before the multi-column text the command

```
\begin{multicols}{column number}
```

is written in T_EX-mode. The column number is a number in the range of 2-10. Before the formula the multi-column text is ended by inserting the command

```
\end{multicols}
```

⁴⁶see sec. 11.2

⁴⁷**multicol** is part of every L^AT_EX standard installation.

in T_EX-mode.

Due to the command `some` space is automatically added before the formula. To revert this, -6 mm vertical space is inserted before the formula. When the formula style **Indented**⁴⁸ is used, -9 mm space is inserted instead.

As example a multi-column text with a displayed formula:

Das Spektrum wird fouriertransformiert. Die Fouriertransformation wird verwendet, um die überlagerten Signale (Netzwerk, Lösungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des Ausgangssignals. Im Expe-	riment haben wir es mit sehr vielen Teilchen zu tun, so dass man über alle Phasen integrieren muss. Sei nun S unser normiertes Ausgangssignal und P die Phasenverteilungsfunktion, so ergibt sich die Beziehung
---	---

$$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t) e^{i\phi} d\phi \quad (28)$$

wobei S_0 das Signal ohne Gradient ist und die Normierungsbedingung $\int_{-\infty}^{\infty} P(\phi, t) d\phi = 1$ gilt. Nun dürfen wir aber nicht den Relaxationsprozess außer Acht lassen. Direkt nach dem $\pi/2$ -rf-Puls	beginnt sich die Magnetisierung zu entfokussieren, wodurch sich das Signal zusätzlich abschwächt. Diese Abschwächung verläuft exponentiell in Abhängigkeit der so genannten T_2 -Zeit.
---	--

23.8. Formulas with Description of Variables

To describe variables within a formula, like in formula (29), a $2 \times n$ matrix is used with left aligned columns for the n used variables.⁴⁹ To set the description in a smaller size, before the matrix e.g. the command `\footnotesize` is inserted.⁵⁰

When the formula style **Indented**⁵¹ is used, a `\hfill`⁵² is inserted before and after the matrix to have the same separation of the matrix from the equation and the side margin.

When the formula style **Centered** is used, the method described in sec.18.2.3 is used to indent formulas. Formula (29) consists of five columns whereas in the first

⁴⁸formula styles see sec. 17

⁴⁹matrices see sec. 4

⁵⁰font sizes see sec. 11.4

⁵¹formula styles see sec. 17

⁵²`\hfill` only works in formulas with the style **Indented**, see sec. 8.2.

two columns contain the equation, the third the matrix and the last one an empty T_EX-brace.

$$F_A = \rho \cdot V \cdot g \qquad \begin{array}{l} \rho \text{ density} \\ V \text{ volume} \\ g \text{ gravitational acceleration} \end{array} \qquad (29)$$

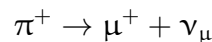
23.9. Upright small Greek Letters

Most of the math fonts only provide italic small Greek letters. But for symbols of elementary particles like pions and neutrinos, upright Greek letters are needed. The file **upgreek.sty**⁵³ that is loaded with the L^AT_EX-preamble line

```
\usepackage{upgreek}
```

provides them. They are created when the command for a small Greek letters is started with **up**. For example the command **\uptau** creates this: τ

With these commands reactions of elementary particles can be typeset:



The upright letters are more bold and wider than the italic ones. They should therefore not be used for units like “ μm ”.

23.10. Text Characters in Formulas

In some cases you might want to insert text characters directly into formulas. When for example the centered dot \cdot is often used in formulas like $\nu = 5 \cdot 10^5 \text{ Hz}$, one would have to insert the command **\cdot**⁵⁴ all the time, because this character is defined in all encodings as text character. But the encoding can be changed by this L^AT_EX-preamble line:

```
\DeclareInputtext{183}{\ifmmode\cdot\else\textperiodcentered\fi}
```

The character encoding (menu Document▷Settings▷Language) specifies what character appears when a keyboard key is pressed. When the key for the character ‘ \cdot ’ is pressed, internally the command **\textperiodcentered** is used. But this command is not available in a formula so that you would get L^AT_EX-errors. With the changed encoding the right command is chosen automatically, depending on if the character was inserted into a formula or not.

The encoding of several characters is saved in definition files. Fore example the encoding **latin9** is defined in the file **latin9.def** that is in the installation folder of

⁵³**upgreek** is part of the L^AT_EX-package **was**.

⁵⁴see sec. 10.3

L^AT_EX. Encodings should only be changed via the L^AT_EX-preamble and not in the definition files. Otherwise own documents could not be edited by other L^AT_EX users working on other computers.

Besides the centered dot, in this document the degree sign ° is defined with the following L^AT_EX-preamble line so that it can directly be inserted to a formula:

```
\DeclareInputtext{176}{\ifmmode^\circ\else\textdegree\fi}
```


A. Typographic Advice

This section is a summary of the most important typographic rules, listed in ISO norms.⁵⁵

- Physical units are *always* set upright⁵⁶, no matter if they appear in italic text: 30 km/h
Between the value and the unit is the smallest space, see sec. 8.1.
This convention is automatically fulfilled when the command `\unittwo` is used. When it is entered to a formula, two boxes appear. In the first one the value is inserted, in the second one the unit, and one gets as above: 30 km/h. Note that `\unittwo` is not a real L^AT_EX command but the command `\unit[value]{unit}`, therefore you cannot use it in T_EX code.
- Percent and perthousand signs are set like physical units: 1,2 ‰ alcohol in blood
- The degree sign follows directly on the value: 15°, but not when it is used in units: 15 °C
- In numbers with more than four digits the smallest space is inserted before every third digit to group them: 18 473 588
- For dimensions like 120×90×40 cm the multiplication sign “×” is used. It is available either via the command `\times` or via the menu **Insert**▷**Special Character**▷**Symbols**.
- Functions with names consisting of several letters are set upright to avoid confusions, see sec. 15.1.
- Indices consisting of several letters, are set upright: E_{kin}
Components of matrices are set italic: \hat{H}_{kl}
- The differentiation/integration operator 'd', the Euler's number 'e' and the imaginary unit 'i' should be set upright, to avoid mixing them up with other variables.
- The character that denotes a FOURIER transformation is inserted either by the command `\mathscr_{\mathbf{F}}` or via the menu **Insert**▷**Special Character**▷**Symbols**▷**Letterlike Symbols**: \mathcal{F}
To be able to use the command `\mathscr`, the L^AT_EX package `mathrsfs` needs to be loaded with the L^AT_EX-preamble line `\usepackage{mathrsfs}`.

⁵⁵This collection was partly taken from the German semi-official dictionary called “Duden” [8] that lists some of the ISO rules. More rules are listed in [9].

⁵⁶done with font styles, see sec. 11.1

B. Synonyms

Some characters and symbols can be created with several commands. Here is a list of the synonym commands:

Command	equivalent to
<code>\ast</code>	<code>*</code>
<code>\choose</code>	<code>\binom</code>
<code>\geq</code>	<code>\ge</code>
<code>\lbrace</code>	<code>{</code>
<code>\lbracket</code>	<code>[</code>
<code>\leftarrow</code>	<code>\gets</code>
<code>\leq</code>	<code>\le</code>
<code>\lor</code>	<code>\vee</code>
<code>\neq</code>	<code>\not=</code>
<code>\slash</code>	<code>/</code>
<code>\vert</code>	<code> </code>

Command	equivalent to
<code>\backslash</code>	<code>\\</code>
<code>\dasharrow</code>	<code>\dashrightarrow</code>
<code>\land</code>	<code>\wedge</code>
<code>\rbrace</code>	<code>}</code>
<code>\rbracket</code>	<code>]</code>
<code>\rightarrow</code>	<code>\to</code>
<code>\lnot</code>	<code>\neg</code>
<code>\ne</code>	<code>\not=</code>
<code>\owns</code>	<code>\ni</code>
<code>\square</code>	<code>\Box</code>
<code>\Vert</code>	<code> </code>

References

- [1] MITTELBACH, F. ; GOOSSENS, M.: *The L^AT_EX Companion*. Addison Wesley, 2004
- [2] [Description](#) of L^AT_EX's math abilities
- [3] [Description](#) of $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX
- [4] [List](#) of all symbols available with L^AT_EX-packages
- [5] [Documentation](#) of the L^AT_EX-package **hyperref**
- [6] [Documentation](#) of the L^AT_EX-package **mhchem**
- [7] [Description](#) of the command `\mathclap`, described in sec. 10.2
- [8] *Duden Band 1*. 22. Auflage, Duden 2001
- [9] [Check list](#) for reviewing manuscripts

Index

- €, 35
- °, 40
- Å, 38
- Accents, 16
 - for one character, 16
 - for operators, 16
 - for several characters, 17
 - in text, 38
- Arrows, 14
 - diagonal, 15
 - horizontal, 15
 - labeled, 15
 - vertical, 15
- Binomial coefficients, 6
- Boxes, 20
 - as paragraph, 23
 - colored, 22
 - with frame, 20
 - without frame, 21
- Bracket size
 - automatic, 12
 - manual, 11
- Brackets, 11
 - for multiline expressions, 43
 - horizontal, 13
 - vertical, 11
- Case differentiations, 6
- Chemical characters
 - Isotopes, 7
 - Symbols, 52
- Chemical equations, 52
- Comma, 60
- Commands
 - @!\@addtoreset, 50
 - @!\@removefromreset, 51
- A
 - \Alph, 52
 - \addtocounter, 49
 - \aligned, 47
 - \alignedat, 47
 - \alph, 52
 - \arabic, 51, 52
 - \arraycolsep, 10, 42
 - \arraystretch, 11
- B
 - \big, 11
 - \bigl - \bigr, 12
 - \bigm, 12
 - \binom, 6
 - \boldmath, 30
 - \boldsymbol, 33
 - \boxed, 20
 - \brace, 6
 - \brack, 6
- C
 - \cases, 6
 - \CD, 54
 - \cdots, 8
 - \ce, 53
 - \cf, 53
 - \cfrac, 5
 - \colorbox, 22
- D
 - \DeclareMathOperator, 28, 37
 - \dbinom, 6
 - \definecolor, 23
 - \dfrac, 4
 - \displaystyle, 1, 32
 - \dotfill, 9
 - \dots, 8
- E
 - \euro, 35
- F
 - \fbox, 20
 - \fcolorbox, 23
 - \frac, 4
 - \framebox, 20
- G
 - \gathered, 47
 - \genfrac, 61

H

`\hdotsfor`, 9
`\hfill`, 19
`\hphantom`, 8
`\hrulefill`, 9
`\hspace`, 19, 43

I

`\int`, 25
`\intertext`, 47

J

`\jot`, 14, 41

L

`\ldots`, 8
`\left`, 10, 12, 44
`\lefteqn`, 43
`\leftroot`, 6
`\lim`, 37
`\linewidth`, 24

M

`\makebox`, 21
`\mathbin`, 29
`\mathclap`, 27, 69
`\mathindent`, 40
`\mathop`, 29
`\mathscr`, 67
`\mathsurround`, 19
`\mbox`, 21
`\middle`, 13
`\multilinegap`, 46

N

`\newcommand`, 55
`\nicefrac`, 5
`\not`, 7
`\numberwithin`, 51

O

`\officialeuro`, 35
`\oldstylenums`, 39
`\overbrace`, 13
`\overline`, 8
`\overset`, 16, 17

P

`\parbox`, 24
`\phantom`, 7
`\prod`, 25

R

`\Roman`, 52
`\raisebox`, 21
`\renewcommand`, 11
`\renewcommand`, 52
`\right`, 10, 12, 44
`\roman`, 52
`\root`, 5
`\rule`, 8

S

`\setlength`, 19
`\shoveleft`, 46
`\shoveright`, 46
`\sideset`, 16
`\smallmatrix`, 11
`\split`, 47
`\sqrt`, 5
`\stackrel`, 36
`\subarray`, 27
`\substack`, 27
`\sum`, 25

T

`\tag`, 50
`\tbinom`, 6
`\texorpdfstring`, 63
`\text`, 2
`\textbackslash`, 11
`\textcircled`, 39
`\textcolor`, 23, 31
`\textvisiblespace`, 3
`\tfrac`, 4

U

`\unboldmath`, 30
`\underbrace`, 13
`\underline`, 8
`\underset`, 16, 17
`\unitfrac`, 5
`\uproot`, 6

V

`\vphantom`, 8, 44

X

`\xleftarrow`, 15
`\xrightarrow`, 15

- Comparisons, *see* Relations
- Cross-references
 - to formulas, 48
- Delimiters, 11
- Diagrams
 - amscd, 54
 - xymatrix, 55
- Ellipses, 8
- Exponents, 4
- Font
 - size, 31
 - style, 29
- Fonts, 29
- Formula
 - bold, 30
 - canceled, 62
 - colored, 31
 - display style, 1
 - in multi-column text, 63
 - in section headings, 62
 - inline, 1
 - long, 42
 - multiline, 41
 - align environment, 44
 - alignat environment, 45
 - Column separation, 42
 - eqnarray environment, 45
 - flalign environment, 45
 - formula parts, 47
 - gather environment, 46
 - Line separation, 41
 - multiline environment, 46
 - text, 47
 - numbering, *see* Formula numbering
 - styles, 40
 - underlined, 8
 - with description of variables, 64
- Formula numbering, 48
 - self-defined delimiters, 50
 - subnumbering, 49
 - user-defined, 50
 - with letters, 51
 - with Roman numbers, 51
- Fractions, 4
 - self-defined, 61
- Frames, *see* Boxes
- Functions
 - modulo-, 38
 - predefined, 36
 - self-defined, 36
- Greek letters, 33
 - big, 33
 - bold, 33
 - small, 33
 - upright, 65
- Indices, 4
- Integrals, 25
- Isotopes, *see* Chemical characters
- \LaTeX -preamble, 2
- Limits, 37
- Lines, 8
- Macros, 56
 - Toolbar, 58
- Mathematical text, 2
- Matrices, 9
- Minuscule numbers, 39
- Negations, 7
- Numbers
 - negative, 60
- Operators, 25
 - big, 25
 - binary, 28
 - Limits, 26
 - self-defined, 28
- Packages
 - braket, 60
 - calc, 24, 41
 - cancel, 62
 - carlisle, 51
 - color, 22
 - eurosym, 35

- hyperref, 62, 69
- icomma, 60
- mathrsfs, 67
- mhchem, 53, 69
- multicol, 63
- remreset, 51
- upgreek, 65
- was, 60, 65

physical, 60

Placeholders, 7

Relations, 35

Roots, 5

Space

- besides inline formulas, 19
- horizontal, 18
 - predefined, 18
 - variable, 19

Special characters, 38

- miscellaneous, 40

Subscripts, *see* Indices

Sums, 25

Superscripts, *see* Exponents

Symbols, 34

- chemical, 52
- Euro-symbol, 35
- mathematical, 34
- miscellaneous, 34

Synonyms, 68

TEX-braces, 2

TEX-mode, 2

Text

- colored, 23
- in formulas, 2, 47, 65

Tilde, 39

Tips, 60

Typographic advice, 67

Umlauts, 16

User-defined commands, 55

- Math macros, 56
- `\newcommand`, 55

Vectors, 16