Type	Inline (within text) formulas	Displayed equations	Displayed and automatically numbered equations
Environment	`math`	`displaymath`	`equation`
LaTeX shorthand	`\(...\)`	`\[...\]`	
TeX shorthand	`$...$`	`$$...$$`	
Comment			`equation*` (starred version) suppresses numbering, but requires amsmath

Suggestion: Using the `$$...$$` should be avoided, as it may cause problems, particularly with the AMS-LaTeX macros. Furthermore, should a problem occur, the error messages may not be helpful.

The `equation*` and `displaymath` environments are functionally equivalent.

If you are typing text normally, you are said to be in *text mode*, but while you are typing within one of those mathematical environments, you are said to be in *math mode*, that has some differences compared to the *text mode*:

1. Most spaces and line breaks do not have any significance, as all spaces are either derived logically from the mathematical expressions, or have to be specified with special commands such as `\quad`
2. Empty lines are not allowed. Only one paragraph per formula.
3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using dedicated commands.[3]

27.1.1. Inserting "Displayed" maths inside blocks of text

In order for some operators, such as `\lim` or `\sum` to be displayed correctly inside some math environments (read `$.....$`), it might be convenient to write the `\displaystyle` class inside the environment. Doing so might cause the line to be taller, but will cause exponents and indices to be displayed correctly for some math operators. For example, the `$\sum$` will print a smaller Σ and `$\displaystyle \sum$` will print a bigger one $\displaystyle\sum$, like in equations (This only works with AMSMATH package). It is also possible to force this behaviour for all math environments by declaring `\everymath{\displaystyle}` at the very beginning (i.e. before `\begin{document}`), which is useful in longer documents.

27.2. Symbols

Mathematics has many symbols! One of the most difficult aspects of learning LaTeX is remembering how to produce symbols. There is of course a set of symbols that can be accessed directly from the keyboard:

3 Chapter 27.11 on page 328

```
+ - = ! / ( ) [ ] < > | ' :
```

Beyond those listed above, distinct commands must be issued in order to display the desired symbols. There are many examples such as Greek letters, set and relations symbols, arrows, binary operators, etc.

For example:

```
\forall x \in X, \quad \exists y \leq \epsilon
```

$$\forall x \in X, \quad \exists y \leq \epsilon$$

Fortunately, there's a tool that can greatly simplify the search for the command for a specific symbol. Look for "Detexify" in the external links[4] section below. Another option would be to look in the "The Comprehensive LaTeX Symbol List" in the external links[5] section below.

27.3. Greek letters

Greek letters are commonly used in mathematics, and they are very easy to type in *math mode*. You just have to type the name of the letter after a backslash: if the first letter is lowercase, you will get a lowercase Greek letter, if the first letter is uppercase (and only the first letter), then you will get an uppercase letter. Note that some uppercase Greek letters look like Latin ones, so they are not provided by LaTeX (e.g. uppercase *Alpha* and *Beta* are just "A" and "B" respectively). Lowercase epsilon, theta, kappa, phi, pi, rho, and sigma are provided in two different versions. The alternate, or *var*iant, version is created by adding "var" before the name of the letter:

```
\alpha, \Alpha, \beta, \Beta, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \mu, \Phi
```

$$\alpha, \mathrm{A}, \beta, \mathrm{B}, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \mu, \Phi$$

Scroll down to #List of Mathematical Symbols[6] for a complete list of Greek symbols.

4 Chapter 27.22 on page 340
5 Chapter 27.22 on page 340
6 Chapter 27.18 on page 338

27.4. Operators

An operator is a function that is written as a word: e.g. trigonometric functions (sin, cos, tan), logarithms and exponentials (log, exp), limits (lim), as well as trace and determinant (tr, det). LaTeX has many of these defined as commands:

```
\cos (2\theta) = \cos^2 \theta - \sin^2 \theta
```

$$\cos(2\theta) = \cos^2\theta - \sin^2\theta$$

For certain operators such as limits[7], the subscript is placed underneath the operator:

```
\lim_{x \to \infty} \exp(-x) = 0
```

$$\lim_{x \to \infty} \exp(-x) = 0$$

For the modular operator[8] there are two commands: `\bmod` and `\pmod`:

```
a \bmod b
```

$$a \bmod b$$

```
x \equiv a \pmod b
```

$$x \equiv a \pmod b$$

To use operators that are not pre-defined, such as argmax[9], see custom operators[10]

7 https://en.wikipedia.org/wiki/Limit%20%28mathematics%29
8 https://en.wikipedia.org/wiki/Modular%20arithmetic
9 https://en.wikipedia.org/wiki/argmax
10 Chapter 28.6 on page 357

27.5. Powers and indices

Powers and indices are equivalent to superscripts and subscripts in normal text mode. The caret (`^`; also known as the circumflex accent[11]) character is used to raise something, and the underscore (`_`) is for lowering. If more than one expression is raised or lowered, they should be grouped using curly braces (`{` and `}`).

```
k_{n+1} = n^2 + k_n^2 - k_{n-1}
```

$$k_{n+1} = n^2 + k_n^2 - k_{n-1}$$

For powers with more than one digit, surround the power with {}.

```
n^{22}
```

$$n^{22}$$

An underscore (`_`) can be used with a vertical bar (|) to denote evaluation using subscript notation in mathematics:

```
f(n) = n^5 + 4n^2 + 2 _{n=17}
```

$$f(n) = n^5 + 4n^2 + 2|_{n=17}$$

27.6. Fractions and Binomials

A fraction is created using the `\frac{numerator}{denominator}` command. (for those who need their memories refreshed, that's the *top* and *bottom* respectively!). Likewise, the binomial coefficient[12] (aka the Choose function) may be written using the `\binom` command[13]:

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
```

11 https://en.wikipedia.org/wiki/Caret
12 https://en.wikipedia.org/wiki/Binomial%20coefficient
13 requires the `amsmath` package

$\frac{n!}{k!(n-k)!} = \binom{n}{k}$

It is also possible to use the `\choose` command without the `amsmath` package:

```
\frac{n!}{k!(n-k)!} = {n \choose k}
```

$\frac{n!}{k!(n-k)!} = \binom{n}{k}$

You can embed fractions within fractions:

```
\frac{\frac{1}{x}+\frac{1}{y}<!---->}{y-z}
```

$\frac{\frac{1}{x}+\frac{1}{y}}{y-z}$

Note that when appearing inside another fraction, or in inline text $\frac{a}{b}$, a fraction is noticeably smaller than in displayed mathematics. The `\tfrac` and `\dfrac` commands[14] force the use of the respective styles, `\textstyle` and `\displaystyle`. Similarly, the `\tbinom` and `\dbinom` commands typeset the binomial coefficient.

Another way to write fractions is to use the `\over` command without the `amsmath` package:

```
{n! \over k!(n-k)!} = {n \choose k}
```

$\frac{n!}{k!(n-k)!} = \binom{n}{k}$

For relatively simple fractions, especially within the text, it may be more aesthetically pleasing to use powers and indices[15]:

14 requires the `amsmath` package
15 Chapter 27.5 on page 313

```
^3/_7
```

$^3/_7$

If this looks a little "loose" (overspaced), a tightened version can be defined by inserting some negative space

```
%running fraction with slash - requires math mode.
\newcommand*\rfrac[2]

\rfrac{3}{7}
```

$^3/_7$

If you use them throughout the document, usage of `xfrac` package is recommended. This package provides `\sfrac` command to create slanted fractions. Usage:

```
Take $\sfrac{1}{2}$ cup of sugar, \dots

  3\times\sfrac{1}{2}=1\sfrac{1}{2}

Take ${}^1/_2$ cup of sugar, \dots

  3\times{}^1/_2=1{}^1/_2
```

Take ½ cup of sugar, …

$$3 \times {}^1/_2 = 1{}^1/_2$$

Take $^1/_2$ cup of sugar, …

$$3 \times {}^1/_2 = 1{}^1/_2$$

Figure 81

If fractions are used as an exponent curly braces have to be used around the `\sfrac` command:

$x^\frac{1}{2}$ % no error $x^\sfrac{1}{2}$ % error $x^{\sfrac{1}{2}}$ % no error

```
$x^\frac{1}{2}$ % no error
```

$x^{\frac{1}{2}}$

In some cases, using the package alone will result in errors about certain font shapes not being available. In that case, the `lmodern` and `fix-cm` packages need to be added as well.

Alternatively, the `nicefrac` package provides the `\nicefrac` command, whose usage is similar to `\sfrac`.

27.6.1. Continued fractions

Continued fractions should be written using `\cfrac` command[16]:

```
\begin{equation}
  x = a_0 + \cfrac{1}{a_1
          + \cfrac{1}{a_2
          + \cfrac{1}{a_3 + \cfrac{1}{a_4} } } }
\end{equation}
```

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$

27.6.2. Multiplication of two numbers

To make multiplication visually similar to a fraction, a nested array can be used, for example multiplication of numbers written one below the other.

16 requires the `amsmath` package

```
\begin{equation}
\frac{
    \begin{array}[b]{r}
      \left( x_1 x_2 \right)\\
      \times \left( x'_1 x'_2 \right)
    \end{array}
  }{
    \left( y_1y_2y_3y_4 \right)
  }
\end{equation}
```

$$\frac{\begin{array}[b]{r}\left(x_1 x_2 \right)\\ \times \left(x'_1 x'_2 \right)\end{array}}{\left(y_1y_2y_3y_4 \right)}$$

27.7. Roots

The `\sqrt` command creates a square root surrounding an expression. It accepts an optional argument specified in square brackets (`[` and `]`) to change magnitude:

```
\sqrt{\frac{a}{b}}
```

$$\sqrt{\frac{a}{b}}$$

```
\sqrt[n]{1+x+x^2+x^3+\ldots}
```

$$\sqrt[n]{1+x+x^2+x^3+\ldots}$$

Some people prefer writing the square root "closing" it over its content. This method arguably makes it more clear what is in the scope of the root sign. This habit is not normally used while writing with the computer, but if you still want to change the output of the square root, LaTeX gives you this possibility. Just add the following code in the preamble of your document:

```
% New definition of square root:
% it renames \sqrt as \oldsqrt
\let\oldsqrt\sqrt
% it defines the new \sqrt in terms of the old one
\def\sqrt{\mathpalette\DHLhksqrt}
\def\DHLhksqrt#1#2{%
\setbox0=\hbox{$#1\oldsqrt{#2\,}$}\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}
```

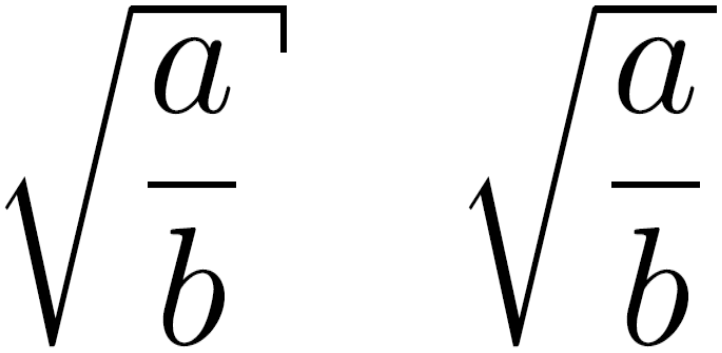


Figure 82 The new style is on left, the old one on right

This TeX code first renames the `\sqrt` command as `\oldsqrt`, then redefines `\sqrt` in terms of the old one, adding something more. The new square root can be seen in the picture on the left, compared to the old one on the right. Unfortunately this code won't work if you want to use multiple roots: if you try to write $\sqrt[b]{a}$ as `\sqrt[b]{a}` after you used the code above, you'll just get a wrong output. In other words, you can redefine the square root this way only if you are not going to use multiple roots in the whole document.

An alternative piece of TeX code that does allow multiple roots is

```
\usepackage{letltxmacro}
\makeatletter
\let\oldr@@t\r@@t
\def\r@@t#1#2{%
\setbox0=\hbox{$\oldr@@t#1{#2\,}$}\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}
\LetLtxMacro{\oldsqrt}{\sqrt}
\renewcommand*{\sqrt}[2][\ ]{\oldsqrt[#1]{#2} }
\makeatother


$\sqrt[a]{b} \quad \oldsqrt[a]{b}$
```

Figure 83

However this requires the `\usepackage{letltxmacro}` package

27.8. Sums and integrals

The `\sum` and `\int` commands insert the sum and integral symbols respectively, with limits specified using the caret (`^`) and underscore (`_`). The typical notation for sums is:

```
\sum_{i=1}^{10} t_i
```

$\sum_{i=1}^{10} t_i$

or

```
\displaystyle\sum_{i=1}^{10} t_i
```

$$\sum_{i=1}^{10} t_i$$

The limits for the integrals follow the same notation. It's also important to represent the integration variables with an upright d, which in math mode is obtained through the \mathrm{} command, and with a small space separating it from the integrand, which is attained with the \, command.

```
\int_0^\infty \mathrm{e}^{-x}\,\mathrm{d}x
```

$\int_0^\infty \mathrm{e}^{-x}\,\mathrm{d}x$

There are many other "big" commands which operate in a similar manner:

`\sum`	$\sum$	`\prod`	$\prod$	`\coprod`	
`\bigoplus`	$\bigoplus$	`\bigotimes`	$\bigotimes$	`\bigodot`	$\bigodot$
`\bigcup`	$\bigcup$	`\bigcap`	$\bigcap$	`\biguplus`	$\biguplus$
`\bigsqcup`	$\bigsqcup$	`\bigvee`	$\bigvee$	`\bigwedge`	$\bigwedge$
`\int`	$\int$	`\oint`	$\oint$	`\iint`[17]	$\iint$
`\iiint`[18]	$\iiint$	`\iiiint`[19]	$\iiiint$	`\idotsint`[20]	$\int\cdots\int$

For more integral symbols, including those not included by default in the Computer Modern font, try the `esint` package.

The `\substack` command[21] allows the use of `\\` to write the limits over multiple lines:

```
\sum_{\substack{
  0<i<m \\
  0<j<n
 }<!---->}
P(i,j)
```

$$\sum_{\substack{0<i<m\\0<j<n}} P(i,j)$$

If you want the limits of an integral to be specified above and below the symbol (like the sum), use the `\limits` command:

```
\int\limits_a^b
```

$$\int\limits_a^b$$

However if you want this to apply to ALL integrals, it is preferable to specify the `intlimits` option when loading the `amsmath` package:

```
\usepackage[intlimits]{amsmath}
```

17 requires the `amsmath` package
18 requires the `amsmath` package
19 requires the `amsmath` package
20 requires the `amsmath` package
21 requires the `amsmath` package

Subscripts and superscripts in other contexts as well as other parameters to `amsmath` package related to them are described in Advanced Mathematics[22] chapter.

For bigger integrals, you may use personal declarations, or the `bigints` package [23].

27.9. Brackets, braces and delimiters

How to use braces in multi line equations is described in the Advanced Mathematics[24] chapter.

The use of delimiters such as brackets soon becomes important when dealing with anything but the most trivial equations. Without them, formulas can become ambiguous. Also, special types of mathematical structures, such as matrices, typically rely on delimiters to enclose them.

There are a variety of delimiters available for use in LaTeX:

```
( a ), [ b ], \{ c \},  d , \ e \,
\langle f \rangle, \lfloor g \rfloor,
\lceil h \rceil, \ulcorner i \urcorner
```

$(a), [b], \{c\}, |d|, \|e\|, \langle f \rangle, \lfloor g \rfloor, \lceil h \rceil, \ulcorner i \urcorner$

where \lbrack and \rbrack may be used in place of [and].

27.9.1. Automatic sizing

Very often mathematical features will differ in size, in which case the delimiters surrounding the expression should vary accordingly. This can be done automatically using the `\left`, `\right`, and `\middle` commands. Any of the previous delimiters may be used in combination with these:

```
\left(\frac{x^2}{y^3}\right)
```

$\left(\frac{x^2}{y^3}\right)$

22 Chapter 28.7 on page 357

23 `http://hdl.handle.net/2268/6219`

24 Chapter 28.2.3 on page 348

```
P\left(A=2\middle\frac{A^2}{B}>4\right)
```

$$P\left(A=2\middle|\frac{A^2}{B}>4\right)$$

Figure 84

Curly braces are defined differently by using `\left\{` and `\right\}`,

```
\left\{\frac{x^2}{y^3}\right\}
```

$$\left\{\frac{x^2}{y^3}\right\}$$

If a delimiter on only one side of an expression is required, then an invisible delimiter on the other side may be denoted using a period (`.`).

```
\left.\frac{x^3}{3}\right_0^1
```

$$\frac{x^3}{3}\Big|_0^1$$

27.9.2. Manual sizing

In certain cases, the sizing produced by the `\left` and `\right` commands may not be desirable, or you may simply want finer control over the delimiter sizes. In this case, the `\big`, `\Big`, `\bigg` and `\Bigg` modifier commands may be used:

```
( \big( \Big( \bigg( \Bigg(
```

$$(\big(\Big(\bigg(\Bigg($$

These commands are primarily useful when dealing with nested delimiters. For example, when typesetting

```
\frac{\mathrm d}{\mathrm d x} \left( k g(x) \right)
```

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(kg(x)\right)$$

we notice that the `\left` and `\right` commands produce the same size delimiters as those nested within it. This can be difficult to read. To fix this, we write

```
\frac{\mathrm d}{\mathrm d x} \big( k g(x) \big)
```

$$\frac{\mathrm{d}}{\mathrm{d}x}\big(kg(x)\big)$$

Manual sizing can also be useful when an equation is too large, trails off the end of the page, and must be separated into two lines using an align command. Although the commands `\left.` and `\right.` can be used to balance the delimiters on each line, this may lead to wrong delimiter sizes. Furthermore, manual sizing can be used to avoid overly large delimiters if an `\underbrace` or a similar command appears between the delimiters.

27.9.3. Typesetting intervals

To denote open and half-open intervals, the notations]a,b[, (a,b),]a,b], (a,b], [a,b[and [a,b) are used. If the square bracket notation is used, then the interval must be put between curly braces (`{` and `}`) in order to have correct spacing. Similarly, if a (half-)open interval starts with a negative number, then the number including its minus-symbol must also be put between curly brackets, so that LaTeX understands that the minus-symbol is the unary operation. Compare:

```
x \in [-1,1]
```

$$x \in [-1,1]$$

```
x \in {[-1,1]}
```

$x \in [-1, 1]$

```
x \in {[{-1},1]}
```

$x \in [-1, 1]$

27.10. Matrices and arrays

A basic matrix may be created using the `matrix` environment[25]: in common with other table-like structures, entries are specified by row, with columns separated using an ampersand (`&`) and a new rows separated with a double backslash (`\\`)

```
\begin{matrix}
 a & b & c \\
 d & e & f \\
 g & h & i
\end{matrix}
```

$$\begin{matrix}
 a & b & c \\
 d & e & f \\
 g & h & i
\end{matrix}$$

To specify alignment of columns in the table, use starred version[26]:

```
\begin{matrix}
 -1 & 3 \\
 2 & -4
\end{matrix}
=
\begin{matrix*}[r]
 -1 & 3 \\
 2 & -4
\end{matrix*}
```

$$\begin{matrix}
 -1 & 3 \\
 2 & -4
\end{matrix}
=
\begin{matrix*}[r]
 -1 & 3 \\
 2 & -4
\end{matrix*}$$

25 requires the `amsmath` package
26 requires the `mathtools` package

The alignment by default is `c` but it can be any column type valid in `array` environment.

However matrices are usually enclosed in delimiters of some kind, and while it is possible to use the `\left` and `\right` commands[27], there are various other predefined environments which automatically include delimiters:

Environment name	**Surrounding delimiter**	**Notes**
`pmatrix`[28]	( )	centers columns by default
`pmatrix*`[29]	( )	allows to specify alignment of columns in optional parameter
`bmatrix`[30]	[ ]	centers columns by default
`bmatrix*`[31]	[ ]	allows to specify alignment of columns in optional parameter
`Bmatrix`[32]	{ }	centers columns by default
`Bmatrix*`[33]	{ }	allows to specify alignment of columns in optional parameter
`vmatrix`[34]	| |	centers columns by default
`vmatrix*`[35]	| |	allows to specify alignment of columns in optional parameter
`Vmatrix`[36]	‖ ‖	centers columns by default
`Vmatrix*`[37]	‖ ‖	allows to specify alignment of colums in optional parameter

When writing down arbitrary sized matrices, it is common to use horizontal, vertical and diagonal triplets of dots (known as ellipses[38]) to fill in certain columns and rows. These can be specified using the `\cdots`, `\vdots` and `\ddots` respectively:

27 Chapter 27.9.1 on page 321
28 requires the `amsmath` package
29 requires the `mathtools` package
30 requires the `amsmath` package
31 requires the `mathtools` package
32 requires the `amsmath` package
33 requires the `mathtools` package
34 requires the `amsmath` package
35 requires the `mathtools` package
36 requires the `amsmath` package
37 requires the `mathtools` package
38 `https://en.wikipedia.org/wiki/ellipsis`

```
A_{m,n} =
\begin{pmatrix}
 a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
 a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
 \vdots  & \vdots  & \ddots & \vdots  \\
 a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

In some cases you may want to have finer control of the alignment within each column, or want to insert lines between columns or rows. This can be achieved using the `array` environment, which is essentially a math-mode version of the `tabular` environment[39], which requires that the columns be pre-specified:

```
\begin{array}{cc}
 1 & 2 \\
 \hline
 3 & 4
\end{array}
```

$$\begin{array}{cc} 1 & 2 \\ \hline 3 & 4 \end{array}$$

You may see that the AMS matrix class of environments doesn't leave enough space when used together with fractions resulting in output similar to this:

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

To counteract this problem, add additional leading space with the optional parameter to the \\ command:

39 Chapter 14.6 on page 171

```
M = \begin{bmatrix}
      \frac{5}{6} & \frac{1}{6} & 0           \\[0.3em]
      \frac{5}{6} & 0           & \frac{1}{6} \\[0.3em]
      0           & \frac{5}{6} & \frac{1}{6}
    \end{bmatrix}
```

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

If you need ”border” or ”indexes” on your matrix, plain TeX provides the macro `\bordermatrix`

```
M = \bordermatrix{~ & x & y \cr
                  A & 1 & 0 \cr
                  B & 0 & 1 \cr}
```

$$M = \bordermatrix{~ & x & y \cr A & 1 & 0 \cr B & 0 & 1 \cr}$$

Figure 85

27.10.1. Matrices in running text

To insert a small matrix, and not increase leading in the line containing it, use `smallmatrix` environment:

```
A matrix in text must be set smaller:
$\bigl(\begin{smallmatrix}
a&b \\ c&d
\end{smallmatrix} \bigr)$
to not increase leading in a portion of text.
```

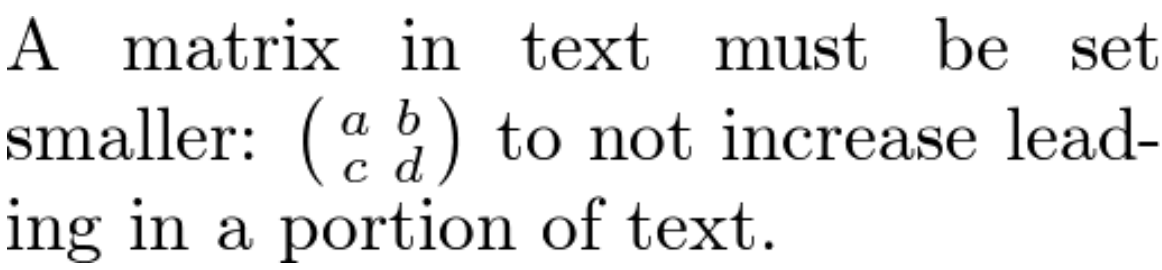


Figure 86

27.11. Adding text to equations

The math environment differs from the text environment in the representation of text. Here is an example of trying to represent text within the math environment:

```
50 apples \times 100 apples = lots of apples^2
```

$$50apples \times 100apples = lotsofapples^2$$

There are two noticeable problems: there are no spaces between words or numbers, and the letters are italicized and more spaced out than normal. Both issues are simply artifacts of the maths mode, in that it treats it as a mathematical expression: spaces are ignored (LaTeX spaces mathematics according to its own rules), and each character is a separate element (so are not positioned as closely as normal text).

There are a number of ways that text can be added properly. The typical way is to wrap the text with the `\text{...}` command [40] (a similar command is `\mbox{...}`, though this causes problems with subscripts, and has a less descriptive name). Let's see what happens when the above equation code is adapted:

```
50 \text{apples} \times 100 \text{apples}
= \text{lots of apples}^2
```

$$50\text{apples} \times 100\text{apples} = \text{lots of apples}^2$$

The text looks better. However, there are no gaps between the numbers and the words. Unfortunately, you are required to explicitly add these. There are many ways to add spaces

40 requires the `amsmath` package

between maths elements, but for the sake of simplicity we may simply insert space characters into the `\text` commands.

```
50 \text{ apples} \times 100 \text{ apples}
= \text{lots of apples}^2
```

$$50 \text{ apples} \times 100 \text{ apples} = \text{lots of apples}^2$$

27.11.1. Formatted text

Using the `\text` is fine and gets the basic result. Yet, there is an alternative that offers a little more flexibility. You may recall the introduction of font formatting commands[41], such as `\textrm`, `\textit`, `\textbf`, etc. These commands format the argument accordingly, e.g., `\textbf{bold text}` gives **bold text**. These commands are equally valid within a maths environment to include text. The added benefit here is that you can have better control over the font formatting, rather than the standard text achieved with `\text`.

```
50 \textrm{ apples} \times 100
\textbf{ apples} = \textit{lots of apples}^2
```

$$50 \text{ apples} \times 100 \textbf{ apples} = \textit{lots of apples}^2$$

27.12. Formatting mathematics symbols

See also: w:Mathematical Alphanumeric Symbols[42], w:Help:Displaying a formula#Alphabets and typefaces[43] and w:Wikipedia:LaTeX symbols#Fonts[44]

We can now format text; what about formatting mathematical expressions? There are a set of formatting commands very similar to the font formatting ones just used, except that they are specifically aimed at text in math mode (requires `amsfonts`)

LaTeX command	Sample	Description	Common use
`\mathnormal{…}` (or simply omit any command)	$ABCDEF\ abcdef\ 123456$	The default math font	Most mathematical notation
`\mathrm{…}`	$\mathrm{ABCDEF\ abcdef\ 123456}$	This is the default or normal font, unitalicised	Units of measurement, one word functions
`\mathit{…}`	$\mathit{ABCDEF\ abcdef\ 123456}$	Italicised font	Multi-letter function or variable names. Compared to `\mathnormal`, words are spaced more naturally and numbers are italicized as well.

41 Chapter 9.6 on page 99

42 `https://en.wikipedia.org/wiki/Mathematical%20Alphanumeric%20Symbols`

43 `https://en.wikipedia.org/wiki/Help%3ADisplaying%20a%20formula%23Alphabets%20and%20typefaces`

44 `https://en.wikipedia.org/wiki/Wikipedia%3ALaTeX%20symbols%23Fonts`

LaTeX command	Sample	Description	Common use
\mathbf{…}	**ABCDEF abcdef 123456**	Bold font	Vectors
\mathsf{…}	ABCDEF abcdef 123456	Sans-serif[45]	
\mathtt{…}	ABCDEF abcdef 123456	Monospace (fixed-width) font[46]	
\mathfrak{…}	𝔄𝔅ℭ𝔇𝔈𝔉 𝔞𝔟𝔠𝔡𝔢𝔣 𝔞𝔟𝔠𝔡𝔢𝔣	Fraktur[47]	Almost canonical font for Lie algebras, with superscript used to denote New Testament papyri[48], ideals[49] in ring theory
\mathcal{…}	$\mathcal{ABCDEF}$	Calligraphy (uppercase only)	Often used for sheaves/schemes and categories, used to denote cryptological[50] concepts like an *alphabet of definition* ($\mathcal{A}$), *message space* ($\mathcal{M}$), *ciphertext space* ($\mathcal{C}$) and *key space*[51] ($\mathcal{K}$); Kleene's $\mathcal{O}$[52]; naming convention in description logic[53]; Laplace transform[54] ($\mathcal{L}$) and Fourier transform[55] ($\mathcal{F}$)
\mathbb{…} (requires the amsfonts or amssymb package)	𝔸𝔹ℂ𝔻𝔼𝔽	Blackboard bold[56] (uppercase only)	Used to denote special sets (e.g. real numbers)
\mathscr{…} (requires the mathrsfs package)	$\mathscr{ABCDEF}$	Script[57] (uppercase only)	An alternative font for categories and sheaves.

Figure 87

These formatting commands can be wrapped around the entire equation, and not just on the textual elements: they only format letters, numbers, and uppercase Greek, and other math commands are unaffected.

To bold lowercase Greek or other symbols use the `\boldsymbol` command[58]; this will only work if there exists a bold version of the symbol in the current font. As a last resort there is the `\pmb` command[59] (poor mans bold): this prints multiple versions of the character slightly offset against each other.

```
\boldsymbol{\beta} = (\beta_1,\beta_2,\dotsc,\beta_n)
```

$$\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$$

To change the size of the fonts in math mode, see Changing font size[60].

45 https://en.wikipedia.org/wiki/sans-serif
46 https://en.wikipedia.org/wiki/Monospace%20font
47 https://en.wikipedia.org/wiki/Fraktur%20%28script%29
48 https://en.wikipedia.org/wiki/List%20of%20New%20Testament%20papyri
49 https://en.wikipedia.org/wiki/Ideal%20%28ring%20theory%29
50 https://en.wikipedia.org/wiki/Cryptography
51 https://en.wikipedia.org/wiki/key%20space
52 https://en.wikipedia.org/wiki/Kleene%27s%20O
53 https://en.wikipedia.org/wiki/Description%20logic%23Naming%20Convention
54 https://en.wikipedia.org/wiki/Laplace%20transform
55 https://en.wikipedia.org/wiki/Fourier%20transform
56 https://en.wikipedia.org/wiki/Blackboard%20bold
57 https://en.wikipedia.org/wiki/Script%20%28typefaces%29
58 requires the amsmath package
59 requires the amsmath package
60 Chapter 28.9 on page 361

27.12.1. Accents

So what to do when you run out of symbols and fonts? Well the next step is to use accents:

`a' or a^{\prime}`	a'	`a''`	a''
`\hat{a}`	$\hat{a}$	`\bar{a}`	$\bar{a}$
`\grave{a}`	$\grave{a}$	`\acute{a}`	$\acute{a}$
`\dot{a}`	$\dot{a}$	`\ddot{a}`	$\ddot{a}$
`\not{a}`	$\not{a}$	`\mathring{a}`	
`\overrightarrow{AB}`	$\overrightarrow{AB}$	`\overleftarrow{AB}`	$\overleftarrow{AB}$
`a'''`	a'''	`a''''`	a''''
`\overline{aaa}`	$\overline{aaa}$	`\check{a}`	$\check{a}$
`\breve{a}`	$\breve{a}$	`\vec{a}`	$\vec{a}$
`\dddot{a}`[61]		`\ddddot{a}`[62]	
`\widehat{AAA}`	$\widehat{AAA}$	`\widetilde{AAA}`	$\widetilde{AAA}$
`\widehat{AAA}`	$\widehat{AAA}$	`\stackrel\frown{AAA}`	$\overset{\frown}{AAA}$
`\tilde{a}`	$\tilde{a}$	`\underline{a}`	$\underline{a}$

27.13. Color

The package `xcolor`, described in Colors[63], allows us to add color to our equations. For example,

```
k = {\color{red}x} \mathbin{\color{blue}-} 2
```

$$k = {\color{red}x} \mathbin{\color{blue}-} 2$$

The only problem is that this disrupts the default LaTeX formatting around the - operator. To fix this, we enclose it in a `\mathbin` environment, since - is a binary operator. This process is described here[64].

27.14. Plus and minus signs

LaTeX deals with the + and − signs in two possible ways. The most common is as a binary operator. When two maths elements appear on either side of the sign, it is assumed to be a binary operator, and as such, allocates some space either side of the sign. The alternative

61 requires the `amsmath` package
62 requires the `amsmath` package
63 Chapter 8.1 on page 87
64 `http://tex.stackexchange.com/questions/21598/how-to-color-math-symbols`

way is a sign designation. This is when you state whether a mathematical quantity is either positive or negative. This is common for the latter, as in maths, such elements are assumed to be positive unless a − is prefixed to it. In this instance, you want the sign to appear close to the appropriate element to show their association. If you put a + or a − with nothing before it but you want it to be handled like a binary operator you can add an *invisible* character before the operator using `{}`. This can be useful if you are writing multiple-line formulas, and a new line could start with a = or a +, for example, then you can fix some strange alignments adding the invisible character where necessary.

A plus-minus sign is written as:

```
\pm
```

$\pm$

Similarly, there exists also a minus-plus sign:

```
\mp
```

$\mp$

27.15. Controlling horizontal spacing

LaTeX is obviously pretty good at typesetting maths—it was one of the chief aims of the core TeX system that LaTeX extends. However, it can't always be relied upon to accurately interpret formulas in the way you did. It has to make certain assumptions when there are ambiguous expressions. The result tends to be slightly incorrect horizontal spacing. In these events, the output is still satisfactory, yet any perfectionists will no doubt wish to *fine-tune* their formulas to ensure spacing is correct. These are generally very subtle adjustments.

There are other occasions where LaTeX has done its job correctly, but you just want to add some space, maybe to add a comment of some kind. For example, in the following equation, it is preferable to ensure there is a decent amount of space between the maths and the text.

```
\[ f(n) =
  \begin{cases}
    n/2       & \quad \text{if } n \text{ is even}\\
    -(n+1)/2  & \quad \text{if } n \text{ is odd}\\
  \end{cases}
\]
```

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$$

This code produces errors with Miktex 2.9 and does not yield the results seen on the right. Use \mathrm instead of just \text.

(Note that this particular example can be expressed in more elegant code by the `cases` construct provided by the `amsmath` package described in Advanced Mathematics[65] chapter.)

LaTeX has defined two commands that can be used anywhere in documents (not just maths) to insert some horizontal space. They are `\quad` and `\qquad`

A `\quad` is a space equal to the current font size. So, if you are using an 11pt font, then the space provided by `\quad` will also be 11pt (horizontally, of course.) The `\qquad` gives twice that amount. As you can see from the code from the above example, `\quad`s were used to add some separation between the maths and the text.

OK, so back to the fine tuning as mentioned at the beginning of the document. A good example would be displaying the simple equation for the indefinite integral of y with respect to x:

$\int y \, \mathrm{d}x$

If you were to try this, you may write:

```
\int y \mathrm{d}x
```

$\int y\mathrm{d}x$

However, this doesn't give the correct result. LaTeX doesn't respect the white-space left in the code to signify that the y and the $\mathrm{d}x$ are independent entities. Instead, it lumps them altogether. A `\quad` would clearly be overkill in this situation—what is needed are some small spaces to be utilized in this type of instance, and that's what LaTeX provides:

Command	Description	Size
`\,`	small space	3/18 of a quad

65 Chapter 28.2.2 on page 347

Command	Description	Size
\:	medium space	4/18 of a quad
\;	large space	5/18 of a quad
\!	negative space	-3/18 of a quad

NB you can use more than one command in a sequence to achieve a greater space if necessary.

So, to rectify the current problem:

```
\int y\, \mathrm{d}x
```

$$\int y\,\mathrm{d}x$$

```
\int y\: \mathrm{d}x
```

$$\int y\mathrm{d}x$$

```
\int y\; \mathrm{d}x
```

$$\int y\mathrm{d}x$$

The negative space may seem like an odd thing to use, however, it wouldn't be there if it didn't have *some* use! Take the following example:

```
\left(
  \begin{array}{c}
    n \\
    r
  \end{array}
\right) = \frac{n!}{r!(n-r)!}
```

$$\left(\begin{array}{c} n \\ r \end{array}\right) = \frac{n!}{r!(n-r)!}$$

The matrix-like expression for representing binomial coefficients is too padded. There is too much space between the brackets and the actual contents within. This can easily be corrected by adding a few negative spaces after the left bracket and before the right bracket.

```
\left(\!
  \begin{array}{c}
    n \\
    r
  \end{array}
\!\right) = \frac{n!}{r!(n-r)!}
```

$$\left(\!\begin{array}{c} n \\ r \end{array}\!\right) = \frac{n!}{r!(n-r)!}$$

In any case, adding some spaces manually should be avoided whenever possible: it makes the source code more complex and it's against the basic principles of a What You See is What You Mean approach. The best thing to do is to define some commands using all the spaces you want and then, when you use your command, you don't have to add any other space. Later, if you change your mind about the length of the horizontal space, you can easily change it modifying only the command you defined before. Let us use an example: you want the d of a dx in an integral to be in roman font and a small space away from the rest. If you want to type an integral like `\int x \, \mathrm{d} x`, you can define a command like this:

```
\newcommand{\dd}{\mathop{}\,\mathrm{d}}
```

in the preamble of your document. We have chosen `\dd` just because it reminds the "d" it replaces and it is fast to type. Doing so, the code for your integral becomes `\int x \dd x`. Now, whenever you write an integral, you just have to use the `\dd` instead of the "d", and all your integrals will have the same style. If you change your mind, you just have to change the definition in the preamble, and all your integrals will be changed accordingly.

27.16. Manually Specifying Formula Style

To manually display a fragment of a formula using text style, surround the fragment with curly braces and prefix the fragment with `\textstyle`. The braces are required because the `\textstyle` macro changes the state of the renderer, rendering all subsequent mathematics in text style. The braces limit this change of state to just the fragment enclosed within. For example, to use text style for just the summation symbol in a sum, one would enter

```
\begin{equation}
   C^i_j = {\textstyle \sum_k} A^i_k B^k_j
\end{equation}
```

The same thing as a command would look like this:

```
\newcommand{\tsum}[1]}
```

Note the extra braces. Just one set around the expression won't be enough. That would cause all math after `\tsum k` to be displayed using text style.

To display part of a formula using display style, do the same thing, but use `\displaystyle` instead.

27.17. Advanced Mathematics: AMS Math package

The AMS (American Mathematical Society[66]) mathematics package is a powerful package that creates a higher layer of abstraction over mathematical LaTeX language; if you use it it will make your life easier. Some commands `amsmath` introduces will make other plain LaTeX commands obsolete: in order to keep consistency in the final output you'd better use `amsmath` commands whenever possible. If you do so, you will get an elegant output without worrying about alignment and other details, keeping your source code readable. If you want to use it, you have to add this in the preamble:

```
\usepackage{amsmath}
```

27.17.1. Introducing dots in formulas

`amsmath` defines also the `\dots` command, that is a generalization of the existing `\ldots`. You can use `\dots` in both text and math mode and LaTeX will replace it with three dots "..." but it will decide according to the context whether to put it on the bottom (like `\ldots`) or centered (like `\cdots`).

27.17.2. Dots

LaTeX gives you several commands to insert dots (ellipses) in your formulae. This can be particularly useful if you have to type big matrices omitting elements. First of all, here are the main dots-related commands LaTeX provides:

Code	Output	Comment
`\dots`	$\dots$	generic dots (ellipsis), to be used in text (outside formulae as well). It automatically manages whitespaces before and after itself according to the context, it's a higher level command.
`\ldots`	$\dots$	the output is similar to the previous one, but there is no automatic whitespace management; it works at a lower level.
`\cdots`	$\cdots$	These dots are centered relative to the height of a letter. There is also the binary multiplication operator, `\cdot`, mentioned below.
`\vdots`	$\vdots$	vertical dots
`\ddots`	$\ddots$	diagonal dots
`\iddots`		inverse diagonal dots (requires the `mathdots` package)
`\hdotsfor{n}`	.....	to be used in matrices, it creates a row of dots spanning n columns.

66 https://en.wikipedia.org/wiki/American%20Mathematical%20Society

Instead of using `\ldots` and `\cdots`, you should use the semantically oriented commands. It makes it possible to adapt your document to different conventions on the fly, in case (for example) you have to submit it to a publisher who insists on following house tradition in this respect. The default treatment for the various kinds follows American Mathematical Society conventions.

Code	Output	Comment
`A_1,A_2,\dotsc,`	$A_1, A_2, \dots,$ **Figure 88**	for "dots with commas"
`A_1+\dotsb+A_N`	$A_1 + \cdots + A_N$ **Figure 89**	for "dots with binary operators/relations"
`A_1 \dotsm A_N`	$A_1 \cdots A_N$ **Figure 90**	for "multiplication dots"
`\int_a^b \dotsi`	$\int_a^b \dots$ **Figure 91**	for "dots with integrals"
`A_1\dotso A_N`	$A_1 \dots A_N$ **Figure 92**	for "other dots" (none of the above)

27.17.3. Write an equation with the align environment

How to write an equation with the align environment with the `amsmath` package is described in Advanced Mathematics[67].

67 Chapter 28.2.2 on page 347

27.18. List of Mathematical Symbols

All the pre-defined mathematical symbols from the \TeX\ package are listed below. More symbols are available from extra packages.

Relation Symbols

Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
$<$	<	$>$	>	$=$	=	$\parallel$	\parallel	$\nparallel$	\nparallel
$\leq$	\leq	$\geq$	\geq	$\doteq$	\doteq	$\asymp$	\asymp	$\bowtie$	\bowtie
$\ll$	\ll	$\gg$	\gg	$\equiv$	\equiv	$\vdash$	\vdash	$\dashv$	\dashv
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx	$\in$	\in	$\ni$	\ni
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong	$\smile$	\smile	$\frown$	\frown
$\not\subseteq$	\nsubseteq	$\not\supseteq$	\nsupseteq	$\simeq$	\simeq	$\models$	\models	$\notin$	\notin
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\sim$	\sim	$\perp$	\perp	$\mid$	\mid
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\propto$	\propto	$\prec$	\prec	$\succ$	\succ
$\preceq$	\preceq	$\succeq$	\succeq	$\neq$	\neq	$\triangleleft$	\spherical angle	$\measuredangle$	\measuredangle

Binary Operations

Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\bigtriangleup$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\bigtriangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$*$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\bigcirc$	\bigcirc	$\circ$	\circ
$\dagger$	\dagger	$\wedge$	\wedge	$\bullet$	\bullet	$\setminus$	\setminus
$\ddagger$	\ddagger	$\cdot$	\cdot	$\wr$	\wr		\amalg

Set and/or Logic Notation

Symbol	Script	Symbol	Script
$\exists$	\exists	$\rightarrow$	\rightarrow or \to
$\nexists$	\nexists	$\leftarrow$	\leftarrow or \gets
$\forall$	\forall	$\mapsto$	\mapsto
$\neg$	\neg	$\Longrightarrow$	\implies
$\subset$	\subset	$\Rightarrow$	\Rightarrow or \implies
$\supset$	\supset	$\leftrightarrow$	\leftrightarrow
$\in$	\in	$\Longleftrightarrow$	\iff
$\notin$	\notin	$\Leftrightarrow$	\Leftrightarrow (preferred for equivalence (iff))
$\ni$	\ni	$\top$	\top
$\wedge$	\land	$\bot$	\bot
$\vee$	\lor	$\emptyset$ and $\varnothing$	\emptyset and \varnothing

Delimiters

Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
$\vert$	| or \mid (difference in spacing)	$\Vert$	\\|	/	/	\	\backslash
{	\{	}	\}	$\langle$	\langle	$\rangle$	\rangle
$\uparrow$	\uparrow	$\Uparrow$	\Uparrow	$\lceil$	\lceil	$\rceil$	\rceil
$\downarrow$	\downarrow	$\Downarrow$	\Downarrow	$\lfloor$	\lfloor	$\rfloor$	\rfloor

Note: To use the Greek Letters in LaTeX that have the same appearance as their Roman equivalent, just use the Roman form: e.g., A instead of Alpha, B instead of Beta, etc.

Greek Letters

Symbol	Script	Symbol	Script
A and α	A and \alpha	N and ν	N and \nu
B and β	B and \beta	Ξ and ξ	\Xi and \xi
Γ and γ	\Gamma and \gamma	O and o	O and o
Δ and δ	\Delta and \delta	Π, π and ϖ	\Pi, \pi and \varpi
E, ϵ and ε	E, \epsilon and \varepsilon	P, ρ and ϱ	P, \rho and \varrho

Z and ζ	`Z` and `\zeta`		Σ, σ and ς	`\Sigma`, `\sigma` and `\varsigma`
H and η	`H` and `\eta`		T and τ	`T` and `\tau`
Θ, θ and ϑ	`\Theta`, `\theta` and `\vartheta`		Υ and υ	`\Upsilon` and `\upsilon`
I and ι	`I` and `\iota`		Φ, ϕ, and φ	`\Phi`, `\phi` and `\varphi`
K, κ and $\varkappa$	`K`, `\kappa` and `\varkappa`		X and χ	`X` and `\chi`
Λ and λ	`\Lambda` and `\lambda`		Ψ and ψ	`\Psi` and `\psi`
M and μ	`M` and `\mu`		Ω and ω	`\Omega` and `\omega`

Other symbols											
Symbol	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**
∂	`\partial`	$\imath$	`\imath`	$\Re$	`\Re`	∇	`\nabla`	$\aleph$	`\aleph`		
$\eth$	`\eth`		`\jmath`	$\Im$	`\Im`	$\Box$	`\Box`	$\beth$	`\beth`		
$\hbar$	`\hbar`	ℓ	`\ell`	$\wp$	`\wp`	∞	`\infty`	$\gimel$	`\gimel`		

Trigonometric Functions									
Symbol	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**
sin	`\sin`	arcsin	`\arcsin`	sinh	`\sinh`	sec	`\sec`		
cos	`\cos`	arccos	`\arccos`	cosh	`\cosh`	csc	`\csc`		
tan	`\tan`	arctan	`\arctan`	tanh	`\tanh`				
cot	`\cot`		`\arccot`	coth	`\coth`				

If LaTeX does not include a command for the mathematical operator you want to use, for example `\cis` (**c**osine plus **i** times **s**ine), add to your preamble:

```
\DeclareMathOperator\cis{cis}
```

You can then use `\cis` in the document just like `\cos` or any other mathematical operator.

27.19. Summary

As you begin to see, typesetting math can be tricky at times. However, because LaTeX provides so much control, you can get professional quality mathematics typesetting with relatively little effort (once you've had a bit of practice, of course!). It would be possible to keep going and going with math topics because it seems potentially limitless. However, with this tutorial, you should be able to get along sufficiently.

27.20. Notes

27.21. Further reading

- meta:Help:Displaying a formula[68]: Wikimedia uses a subset of LaTeX commands.

68 `https://en.meta.org/wiki/Help%3ADisplaying%20a%20formula`

27.22. External links

- LaTeX maths symbols[69]
- detexify[70]: applet for looking up LaTeX symbols by drawing them
- [ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf `amsmath` documentation]
- LaTeX - The Student Room[71]
- The Comprehensive LaTeX Symbol List[72]
- MathLex - LaTeX math translator and equation builder[73]

pl:LaTeX/Matematyka[74]

69 `http://www.artofproblemsolving.com/Wiki/index.php/LaTeX:Symbols`
70 `http://detexify.kirelabs.org`
71 `http://www.thestudentroom.co.uk/wiki/LaTeX`
72 `http://www.ctan.org/tex-archive/info/symbols/comprehensive`
73 `http://mathlex.org/latex`
74 `https://pl.wikibooks.org/wiki/LaTeX%2FMatematyka`

28. Advanced Mathematics

This page outlines some more advanced uses of mathematics markup using LaTeX. In particular it makes heavy use of the AMS-LaTeX packages supplied by the American Mathematical Society[1].

28.1. Equation numbering

The `equation` environment automatically numbers your equation:

```
\begin{equation}
 f(x)=(x+a)(x+b)
\end{equation}
```

$$f(x) = (x+a)(x+b) \qquad (1)$$

You can also use the `\label` and `\ref` (or `\eqref` from the `amsmath` package) commands to label and reference equations, respectively. For equation number 1, `\ref` results in 1 and `\eqref` results in (1):

```
\begin{equation} \label{eq:someequation}
5^2 - 5 = 20
\end{equation}

this references the equation \ref{eq:someequation}.
```

$$5^2 - 5 = 20 \qquad (1)$$

this references equation 1.

```
\begin{equation} \label{eq:erl}
a = bq + r
\end{equation}

where \eqref{eq:erl} is true if $a$ and $b$ are integers with $b \neq c$.
```

1 https://en.wikipedia.org/wiki/American%20Mathematical%20Society

$$a = bq + r \qquad (1)$$

where (1) is true if a and b
are integers with $b \neq c$.

Further information is provided in the labels and cross-referencing[2] chapter.

To have the enumeration follow from your section or subsection heading, you must use the `amsmath` package or use AMS class documents. Then enter

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

to the preamble to get enumeration at the section level or

```
\numberwithin{equation}{subsection}
```

to have the enumeration go to the subsection level.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
 \numberwithin{equation}{subsection}
 \begin{document}
 \section{First Section}

 \subsection{A subsection}
 \begin{equation}
  L' = {L}{\sqrt{1-\frac{v^2}{c^2}<!-- -->}<!-- -->}
 \end{equation}
\end{document}
```

$$L' = L\sqrt{1 - \frac{v^2}{c^2}} \qquad (1.1.1)$$

If the style you follow requires putting dots after ordinals (as it is required at least in Polish typography), the `\numberwithin{equation}{subsection}` command in the preamble will result in the equation number in the above example being rendered as follows: (1.1..1).

To remove the duplicate dot, add the following command immediately after `\numberwithin{equation}{section}`:

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

For a numbering scheme using `\numberwithin{equation}{subsection}`, use:

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

in the preamble of the document.

Note: Although it may look like the `\renewcommand` works by itself, it won't reset the equation number with each new section. It must be used together with manual equation number resetting after each new section beginning, or with the much cleaner `\numberwithin`.

2 Chapter 21 on page 269

28.1.1. Subordinate equation numbering

To number subordinate equations in a numbered equation environment, place the part of document containing them in a `subequations` environment:

```
\begin{subequations}
Maxwell's equations:
\begin{align}
        B'&=-\nabla \times E,\\
        E'&=\nabla \times B - 4\pi j,
\end{align}
\end{subequations}
```

Maxwell's equations:

$$B' = -\nabla \times E, \qquad (1.1a)$$

$$E' = \nabla \times B - 4\pi j, \qquad (1.1b)$$

Referencing subordinate equations can be done using either of two methods: adding a label after the `\begin{subequations}` command, which will reference the main equation (1.1 above), or adding a label at the end of each line, before the \\ command, which will reference the sub-equation (1.1a or 1.1b above). It is possible to add both labels in case both types of references are needed.

28.2. Vertically aligning displayed mathematics

A problem often encountered with displayed environments (`displaymath` and `equation`) is the lack of any ability to span multiple lines. While it is possible to define lines individually, these will not be aligned.

28.2.1. Above and below

The `\overset` and `\underset` commands[3] typeset symbols above and below expressions. Without AmsTex the same result of `\overset` can be obtained with `\stackrel`. This can be particularly useful for creating new binary relations:

```
\[
 A \overset{!}{=} B; A \stackrel{!}{=} B
\]
```

$$A \overset{!}{=} B; \quad A \overset{!}{=} B$$

3 requires the `amsmath` package

or to show usage of L'Hôpital's rule[4]:

```
\[
 \lim_{x\to 0}{\frac{e^x-1}{2x}<!-- -->}
 \overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{<!-- -->}{=}<!-- -->}
 \lim_{x\to 0}{\frac{e^x}{2}<!-- -->}={\frac{1}{2}<!-- -->}
\]
```

$$\lim_{x\to 0} \frac{e^x-1}{2x} \overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{=}} \lim_{x\to 0} \frac{e^x}{2} = \frac{1}{2}$$

It is convenient to define a new operator that will set the equals sign with H and the provided fraction:

```
\newcommand{\Heq}[1]{\overset{\left[#1\right]}{\underset{\mathrm{H}}{=}}}
```

which reduces the above example to:

```
\[
 \lim_{x\to 0}{\frac{e^x-1}{2x}}
 \Heq{\frac{0}{0}}
 \lim_{x\to 0}{\frac{e^x}{2}}={\frac{1}{2}}
\]
```

If the purpose is to make comments on particular parts of an equation, the `\overbrace` and `\underbrace` commands may be more useful. However, they have a different syntax (and can be aligned with the `\vphantom` command):

```
\[
 z = \overbrace{
   \underbrace{x}_\text{real} + i
   \underbrace{y}_\text{imaginary}
  }^\text{complex number}
\]
```

$$z = \overbrace{\underbrace{x}_{\text{real}} + i \underbrace{y}_{\text{imaginary}}}^{\text{complex number}}$$

Sometimes the comments are longer than the formula being commented on, which can cause spacing problems. These can be removed using the `\mathclap` command[5]:

4 https://en.wikipedia.org/wiki/L%27H%C3%B4pital%27s_rule

5 requires the `mathtools` package

```
\[
 y = a + f(\underbrace{b x}_{
                  \ge 0 \text{ by assumption}<!-- -->})
   = a + f(\underbrace{b x}_{
         \mathclap{\ge 0 \text{ by assumption}<!-- -->}<!-- -->})
\]
```

$$y = a + f(\underbrace{bx}_{\geq 0 \text{ by assumption}}) = a + f(\underbrace{bx}_{\geq 0 \text{ by assumption}})$$

Figure 93

Alternatively, to use brackets instead of braces use `\underbracket` and `\overbracket` commands[6]:

```
\[
 z = \overbracket[3pt]{
     \underbracket{x}_{\text{real}<!---->} +
     \underbracket[0.5pt][7pt]{iy}_{\text{imaginary}<!---->}
     }^{\text{complex number}<!---->}
\]
```

$$z = \overbrace{\underbrace{x}_{\text{real}} + \underbrace{iy}_{\text{imaginary}}}^{\text{complex number}}$$

Figure 94

The optional arguments set the rule thickness and bracket height respectively:

```
\underbracket[rule thickness][bracket height]{argument}_{text below}
```

The `\xleftarrow` and `\xrightarrow` commands[7] produce arrows which extend to the length of the text. Yet again, the syntax is different: the optional argument (using `[` and `]`) specifies the subscript, and the mandatory argument (using `{` and `}`) specifies the superscript (which can be left empty by inserting a blank space).

6 requires the `mathtools` package
7 requires the `amsmath` package

```
\[
 A \xleftarrow{\text{this way}<!-- -->} B
  \xrightarrow[\text{or that way}]{ } C
\]
```

$$A \xleftarrow{\text{this way}<!-- -->} B \xrightarrow[\text{or that way}]{ } C$$

For more extensible arrows, you must use the `mathtools` package:

```
\[
 a \xleftrightarrow[under]{over} b\\
%
 A \xLeftarrow[under]{over} B\\
%
 B \xRightarrow[under]{over} C\\
%
 C \xLeftrightarrow[under]{over} D\\
%
 D \xhookleftarrow[under]{over} E\\
%
 E \xhookrightarrow[under]{over} F\\
%
 F \xmapsto[under]{over} G\\
\]
```

$$a \xleftrightarrow[under]{over} b$$

$$A \xLeftarrow[under]{over} B$$

$$B \xRightarrow[under]{over} C$$

$$C \xLeftrightarrow[under]{over} D$$

$$D \xhookleftarrow[under]{over} E$$

$$E \xhookrightarrow[under]{over} F$$

$$F \xmapsto[under]{over} G$$

Figure 95

and for harpoons:

```
\[
 H \xrightharpoondown[under]{over} I\\
%
 I \xrightharpoonup[under]{over} J\\
%
 J \xleftharpoondown[under]{over} K\\
%
 K \xleftharpoonup[under]{over} L\\
%
 L \xrightleftharpoons[under]{over} M\\
%
 M \xleftrightharpoons[under]{over} N
\]
```

$$H \underset{under}{\overset{over}{\rightharpoondown}} I$$
$$I \underset{under}{\overset{over}{\rightharpoonup}} J$$
$$J \underset{under}{\overset{over}{\leftharpoondown}} K$$
$$K \underset{under}{\overset{over}{\leftharpoonup}} L$$
$$L \underset{under}{\overset{over}{\rightleftharpoons}} M$$
$$M \underset{under}{\overset{over}{\leftrightharpoons}} N$$

Figure 96

28.2.2. `align` and `align*`

The `align` and `align*` environments, available through the `amsmath` package, are used for arranging equations of multiple lines. As with matrices and tables, \\ specifies a line break, and & is used to indicate the point at which the lines should be aligned.

The `align*` environment is used like the `displaymath` or `equation*` environment:

```
\begin{align*}
 f(x) &= (x+a)(x+b) \\
 &= x^2 + (a+b)x + ab
\end{align*}
```

$$\begin{aligned} f(x) &= (x+a)(x+b) \\ &= x^2 + (a+b)x + ab \end{aligned}$$

Note that the `align` environment must not be nested inside an `equation` (or similar) environment. Instead, `align` is a replacement for such environments; the contents inside an `align` are automatically placed in math mode.

`align*` suppresses numbering. To force numbering on a specific line, use the `\tag{...}` command before the line break.

`align` is similar, but automatically numbers each line like the `equation` environment. Individual lines may be referred to by placing a `\label{...}` before the line break. The `\nonumber` or `\notag` command can be used to suppress the number for a given line:

```
\begin{align}
 f(x) &= x^4 + 7x^3 + 2x^2 \nonumber \\
 &\qquad {} + 10x + 12
\end{align}
```

$$\begin{aligned} f(x) &= x^4 + 7x^3 + 2x^2 \\ &\qquad + 10x + 12 \end{aligned} \tag{3}$$

Notice that we've added some indenting on the second line. Also, we need to insert the double braces (`{}`) before the + sign, otherwise latex won't create the correct spacing after the + sign. The reason for this is that without the braces, latex interprets the + sign as a unary operator, instead of the binary operator that it really is.

More complicated alignments are possible, with additional `&`'s on a single line specifying multiple "equation columns", each of which is aligned. The following example illustrates the alignment rule of `align*`:

```
\begin{align*}
 f(x)  &= a x^2+b x +c   &   g(x)  &= d x^3 \\
 f'(x) &= 2 a x +b       &   g'(x) &= 3 d x^2
\end{align*}
```

$$\begin{aligned} f(x) &= ax^2 + bx + c & g(x) &= dx^3 \\ f'(x) &= 2ax + b & g'(x) &= 3dx^2 \end{aligned}$$

28.2.3. Braces spanning multiple lines

If you want a brace to continue across a new line, do the following:

```
\begin{align}
 f(x) &= \pi \left\{ x^4 + 7x^3 + 2x^2 \right.\nonumber\\
 &\qquad \left. {} + 10x + 12 \right\}
\end{align}
```

$$f(x) = \pi \Big\{ x^4 + 7x^3 + 2x^2 \\ + 10x + 12 \Big\} \qquad (4)$$

In this construction, the sizes of the left and right braces are not automatically equal, in spite of the use of `\left\{` and `\right\}`. This is because each line is typeset as a completely separate equation —notice the use of `\right.` and `\left.` so there are no unpaired `\left` and `\right` commands within a line (these aren't needed if the formula is on one line). You can control the size of the braces manually with the `\big`, `\Big`, `\bigg`, and `\Bigg` commands.

Alternatively, the height of the taller equation can be replicated in the other using the `\vphantom` command:

```
\begin{align}
 A &=     \left(\int_t XXX       \right.\nonumber\\
   &\qquad \left.\vphantom{\int_t} YYY \dots \right)
\end{align}
```

$$A = \left(\int_t XXX \\ YYY \dots \right) \qquad (5)$$

Using aligned braces for piecewise functions

You can also use `\left\{` and `\right.` to typeset piecewise functions[8]:

```
\[f(x) = \left\{
  \begin{array}{lr}
    x^2 & : x < 0\\
    x^3 & : x \ge 0
  \end{array}
\right.
\]
```

$$f(x) = \left\{ \begin{array}{lr} x^2 & : x < 0 \\ x^3 & : x \ge 0 \end{array} \right.$$

8 https://en.wikipedia.org/wiki/piecewise%20functions

28.2.4. The cases environment

The `cases` environment[9] allows the writing of piecewise functions:

```
\[
 u(x) =
  \begin{cases}
   \exp{x} & \text{if } x \geq 0 \\
   1       & \text{if } x < 0
  \end{cases}
\]
```

$$u(x) = \begin{cases} \exp x & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

LaTeX will then take care of defining and or aligning the columns.

Within `cases`, text style math is used with results such as:

$$a = \begin{cases} \int x \, \mathrm{d}x \\ b^2 \end{cases}$$

Display style may be used instead, by using the `dcases` environment[10] from `mathtools`:

```
\[
 a =
   \begin{dcases}
     \int x\, \mathrm{d} x\\
     b^2
   \end{dcases}
\]
```

$$a = \begin{cases} \displaystyle\int x \, \mathrm{d}x \\ b^2 \end{cases}$$

Often the second column consists mostly of normal text. To set it in the normal Roman font of the document, the `dcases*` environment may be used:[11]

```
\[
 f(x) = \begin{dcases*}
        x  & when $x$ is even\\
        -x & when $x$ is odd
        \end{dcases*}
\]
```

9 requires the `amsmath` package
10 requires the `mathtools` package
11 requires the `mathtools` package

$$f(x) = \begin{cases} x & \text{when } x \text{ is even} \\ -x & \text{when } x \text{ is odd} \end{cases}$$

28.2.5. Other environments

Although `align` and `align*` are the most useful, there are several other environments that may also be of interest:

Environment name	Description	Notes
`eqnarray` and `eqnarray*`	Similar to `align` and `align*`	Not recommended because spacing is inconsistent
`multline` and `multline*`[12]	First line left aligned, last line right aligned	Equation number aligned vertically with first line and not centered as with other environments
`gather` and `gather*`[13]	Consecutive equations without alignment	
`flalign` and `flalign*`[14]	Similar to `align`, but left aligns first equation column, and right aligns last column	
`alignat` and `alignat*`[15]	Takes an argument specifying number of columns. Allows control of the horizontal space between equations	This environment takes one argument, the number of "equation columns": count the maximum number of `&`s in any row, add 1 and divide by 2. [ftp://ftp.ams.org/ams/doc/amsmath/amsldoc.pdf]

There are also a few environments that don't form a math environment by themselves and can be used as building blocks for more elaborate structures:

Math environment name	Description
`gathered`[16]	Allows gathering equations to be set under each other and assigned a single equation number
`split`[17]	Similar to `align*`, but used inside another displayed mathematics environment
`aligned`[18]	Similar to `align`, to be used inside another mathematics environment.
`alignedat`[19]	Similar to `alignat`, and likewise takes an additional argument specifying the number of columns of equations to set.

For example:

12 requires the `amsmath` package
13 requires the `amsmath` package
14 requires the `amsmath` package
15 requires the `amsmath` package
16 requires the `amsmath` package
17 requires the `amsmath` package
18 requires the `amsmath` package
19 requires the `amsmath` package

```
\begin{equation}
 \left.\begin{aligned}
        B'&=-\partial \times E,\\
        E'&=\partial \times B - 4\pi j,
       \end{aligned}
 \right\}
 \qquad \text{Maxwell's equations}
\end{equation}
```

$$\left.\begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned}\right\} \quad \text{Maxwell's equations} \qquad (1.1)$$

```
\begin{alignat}{2}
 \sigma_1 &= x + y  &\quad \sigma_2 &= \frac{x}{y} \\
 \sigma_1' &= \frac{\partial x + y}{\partial x} & \sigma_2'
    &= \frac{\partial \frac{x}{y}<!---->}{\partial x}
\end{alignat}
```

$$\sigma_1 = x + y \qquad \sigma_2 = \frac{x}{y} \qquad (1)$$

$$\sigma_1' = \frac{\partial x + y}{\partial x} \quad \sigma_2' = \frac{\partial \frac{x}{y}}{\partial x} \qquad (2)$$

```
\begin{gather*}
a_0=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}f(x)\,\mathrm{d}x\\[6pt]
\begin{split}
a_n=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}f(x)\cos nx\,\mathrm{d}x=\\
=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}x^2\cos nx\,\mathrm{d}x
\end{split}\\[6pt]
\begin{split}
b_n=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}f(x)\sin nx\,\mathrm{d}x=\\
=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}x^2\sin nx\,\mathrm{d}x
\end{split}\\[6pt]
\end{gather*}
```

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\,\mathrm{d}x$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx\,\mathrm{d}x =$$

$$= \frac{1}{\pi} \int_{-\pi}^{\pi} x^2 \cos nx\,\mathrm{d}x$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx\,\mathrm{d}x =$$

$$= \frac{1}{\pi} \int_{-\pi}^{\pi} x^2 \sin nx\,\mathrm{d}x$$

Figure 97

28.3. Indented Equations

To indent an equation, you can set `fleqn` in the document class and then specify a certain value for the `\mathindent` variable:

```
\documentclass[a4paper,fleqn]{report}
\usepackage{amsmath}
\setlength{\mathindent}{1cm}
\begin{document}
\noindent Euler's formula is given below:
\begin{equation*}
 e^{ix} = \cos{x} + i \sin{x}.
\end{equation*}
\noindent This is a very important formula.
\end{document}
```

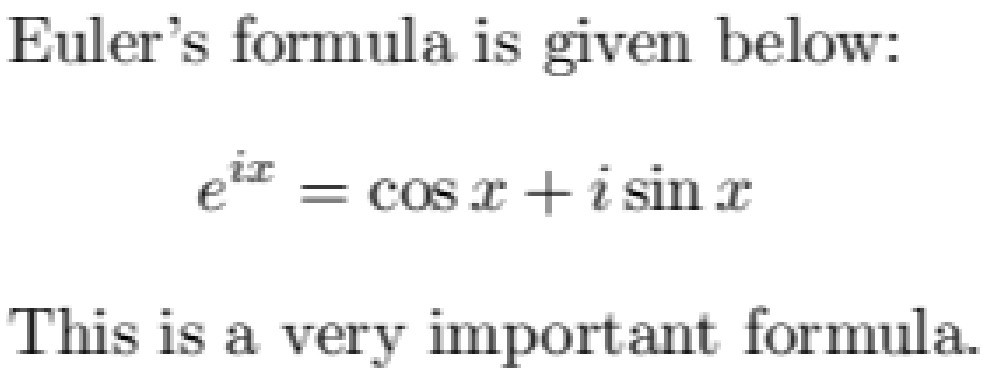


Figure 98

28.4. Page breaks in math environments

To suggest that LaTeX insert a page break inside an `amsmath` environment, you may use the `\displaybreak` command before the line break. Just as with `\pagebreak`, `\displaybreak` can take an optional argument between 0 and 4 denoting the level of desirability of a page break. Whereas 0 means "it is permissible to break here", 4 forces a break. No argument means the same as 4.

Alternatively, you may enable automatic page breaks in math environments with `\allowdisplaybreaks`. It too can have an optional argument denoting the priority of page breaks in equations. Similarly, 1 means "allow page breaks but avoid them" and 4 means "break whenever you want". You can prohibit a page break after a given line using `\\*`.

LaTeX will insert a page break into a long equation if it has additional text added using `\intertext{}` without any additional commands.

Specific usage may look like this:

```
\begin{align*}
 &\vdots\\
 &=12+7 \int_0^2
  \left(
    -\frac{1}{4}\left(e^{-4t_1}+e^{4t_1-8}\right)
  \right)\,dt_1\displaybreak[3]\\
 &= 12-\frac{7}{4}\int_0^2 \left( e^{-4t_1}+e^{4t_1-8} \right)\,dt_1\\
 &\vdots %
\end{align*}
```

$$\vdots$$

$$= 12 + 7 \int_0^2 \left(-\frac{1}{4} \left(e^{-4t_1} + e^{4t_1 - 8} \right) \right) dt_1$$

$$1$$

$$= 12 - \frac{7}{4} \int_0^2 e^{-4t_1} + e^{4t_1 - 8} \, dt_1$$

$$\vdots$$

Figure 99

Page breaks before display maths (of all various forms) are controlled by `\predisplaypenalty`. Its default 10000 means never break immediately before a display. Knuth (*TeXbook* chapter 19) explains this as a printers' tradition not to have a displayed equation at the start of a page. It can be relaxed with

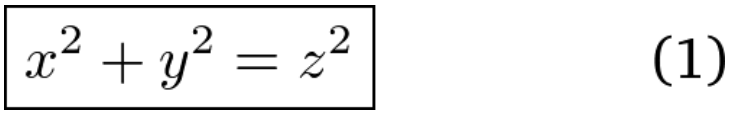
```
\predisplaypenalty=0
```

Sometimes an equation might look best kept together preceding text by a higher penalty, for example a single-line paragraph about a single-line equation, especially at the end of a section.

28.5. Boxed Equations

For a single equation or alignment building block, with the tag outside the box, use `\boxed{}`:

```
\begin{equation}
 \boxed{x^2+y^2 = z^2}
\end{equation}
```

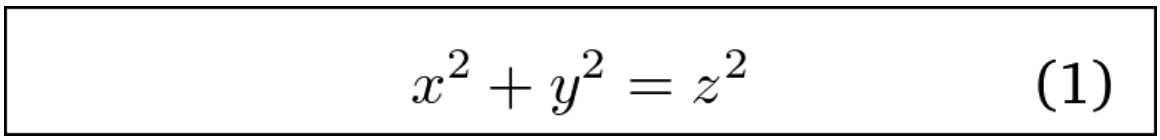


$$x^2 + y^2 = z^2 \qquad (1)$$

Figure 100

If you want the entire line or several equations to be boxed, use a `minipage` inside an `\fbox{}`:

```
\fbox{
 \addtolength{\linewidth}{-2\fboxsep}%
 \addtolength{\linewidth}{-2\fboxrule}%
 \begin{minipage}{\linewidth}
  \begin{equation}
   x^2+y^2=z^2
  \end{equation}
 \end{minipage}
}
```

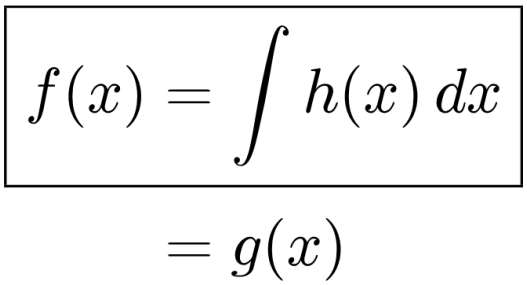


$$x^2 + y^2 = z^2 \qquad (1)$$

Figure 101

There is also the mathtools `\Aboxed{}` which is able to box across alignment marks:

```
\begin{align*}
\Aboxed{ f(x) & = \int h(x)\, dx} \\
             & = g(x)
\end{align*}
```



$$f(x) = \int h(x)\, dx$$

$$= g(x)$$

Figure 102

28.6. Custom operators

Although many common operators[20] are available in LaTeX, sometimes you will need to write your own, e.g. to typeset the argmax[21] operator. The `\operatorname` and `\operatorname*` commands[22] display custom operators; the `*` version sets the underscored option underneath like the `\lim` operator:

```
\[
 \operatorname{arg\,max}_a f(a)
 = \operatorname*{arg\,max}_b f(b)
\]
```

$$\operatorname{arg\,max}_a f(a) = \operatorname*{arg\,max}_b f(b)$$

However, if the operator is frequently used, it is preferable to define a new operator that can be used throughout the entire document. The `\DeclareMathOperator` and `\DeclareMathOperator*` commands[23] are specified in the header of the document:

```
\DeclareMathOperator*{\argmax}{arg\,max}
```

This defines a new command which may be referred to in the body:

```
\[
 \argmax_c f(c)
\]
```

$$\operatorname*{arg\,max}_c f(c)$$

28.7. Advanced formatting

28.7.1. Limits

There are defaults for placement of subscripts and superscripts. For example, limits for the `lim` operator are usually placed below the symbol:

20 Chapter 27.4 on page 312
21 https://en.wikipedia.org/wiki/Arg%20max
22 requires the `amsmath` package
23 requires the `amsmath` package

```
\begin{equation}
  \lim_{a\to \infty} \tfrac{1}{a}
\end{equation}
```

$$\lim_{a\to \infty} \tfrac{1}{a}$$

To override this behavior, use the `\nolimits` operator:

```
\begin{equation}
  \lim\nolimits_{a\to \infty} \tfrac{1}{a}
\end{equation}
```

$$\lim\nolimits_{a\to \infty} \tfrac{1}{a}$$

A `lim` in running text (inside `$...$`) will have its limits placed on the side, so that additional leading won't be required. To override this behavior, use the `\limits` command.

Similarly one can put subscripts under a symbol that usually has them on the side:

```
\begin{equation}
  \int_a^b x^2  \mathrm{d} x
\end{equation}
```

$$\int_a^b x^2 \mathrm{d}x$$

Limits below and under:

```
\begin{equation}
  \int\limits_a^b x^2  \mathrm{d} x
\end{equation}
```

$$\int\limits_a^b x^2 \mathrm{d}x$$

To change the default placement of summation-type symbols to the side for every case, add the `nosumlimits` option to the `amsmath` package. To change the placement for integral symbols, add `intlimits` to the options. `nonamelimits` can be used to change the default for named operators like `det`, `min`, `lim`, etc.

To produce one-sided limits, use `\underset` as follows:

```
\begin{equation}
  \lim_{a \underset{>}{\to} 0} \frac{1}{a}
\end{equation}
```

$$\lim_{a \underset{>}{\to} 0} \frac{1}{a}$$

28.7.2. Subscripts and superscripts

You can place symbols in subscript or superscript (in summation style symbols) with `\nolimits`:

```
\begin{equation}
  \sum\nolimits' C_n
\end{equation}
```

$$\sum\nolimits' C_n$$

It's impossible to mix them with typical usage of such symbols:

```
\begin{equation}
  \sum_{n=1}\nolimits' C_n
\end{equation}
```

$$\sum\nolimits_{n=1}' C_n$$

To add both a prime and a limit to a symbol, one might use the `\sideset` command:

```
\begin{equation}
  \sideset{}{'}\sum_{n=1}C_n
\end{equation}
```

$$\sideset{}{'}\sum_{n=1} C_n$$

It is very flexible: for example, to put letters in each corner of the symbol use this command:

```
\begin{equation}
  \sideset{_a^b}{_c^d}\sum
\end{equation}
```

$$\sum_{a}^{b}\sum_{c}^{d}$$

If you wish to place them on the corners of an arbitrary symbol, you should use `\fourIdx` from the `fouridx` package.

But a simple grouping can also solve the problem:

```
\begin{equation}
  {\sum\limits_{n=1} }'C_n
\end{equation}
```

$$\sum_{n=1}{}'C_n$$

since a math operator can be used with limits or no limits. If you want to change its state, simply group it. You can make it another math operator if you want, and then you can have limits and then limits again.

28.7.3. Multiline subscripts

To produce multiline subscript, use the `\substack` command:

```
\begin{equation}
  \prod_{\substack{
           1\le i \le n\\
           1\le j \le m}<!---->}
     M_{i,j}
\end{equation}
```

$$\prod_{\substack{1\le i\le n\\ 1\le j\le m}} M_{i,j}$$

28.8. Text in aligned math display

To add small interjections in math environments, use the `\intertext` command:

```
\begin{minipage}{3in}
\begin{align*}
\intertext{If}
   A &= \sigma_1+\sigma_2\\
   B &= \rho_1+\rho_2\\
\intertext{then}
C(x) &= e^{Ax^2+\pi}+B
\end{align*}
\end{minipage}
```

If

$$A = \sigma_1 + \sigma_2$$
$$B = \rho_1 + \rho_2$$

then

$$C(x) = e^{Ax^2+\pi} + B$$

Figure 103

Note that any usage of this command does not change the alignment.

Also, in the above example, the command `\shortintertext{}` from the `mathtools` package could have been used instead of intertext to reduce the amount of vertical white space added between the lines.

28.9. Changing font size

There may be a time when you would prefer to have some control over the font size. For example, using text-mode maths, by default a simple fraction will look like this: $\frac{a}{b}$, whereas you may prefer to have it displayed larger, like when in display mode, but still keeping it in-line, like this: $\frac{a}{b}$.

A simple approach is to utilize the predefined sizes for maths elements:

Size command	Description
`\displaystyle`	Size for equations in display mode
`\textstyle`	Size for equations in text mode
`\scriptstyle`	Size for first sub/superscripts
`\scriptscriptstyle`	Size for subsequent sub/superscripts

A classic example to see this in use is typesetting continued fractions (though it's better to use the `\cfrac` command[24] described in the Mathematics[25] chapter instead of the method provided below). The following code provides an example.

```
\begin{equation}
  x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}<!---->}<!---->}
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

As you can see, as the fractions continue, they get smaller (although they will not get any smaller than in this example, where they have reached the `\scriptstyle` limit). If you want to keep the size consistent, you could declare each fraction to use the display style instead; e.g.

```
\begin{equation}
  x = a_0 + \frac{1}{\displaystyle a_1
          + \frac{1}{\displaystyle a_2
          + \frac{1}{\displaystyle a_3 + a_4}<!---->}<!---->}
\end{equation}
```

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + a_4}}}$$

Another approach is to use the `\DeclareMathSizes` command to select your preferred sizes. You can only define sizes for `\displaystyle`, `\textstyle`, etc. One potential downside is that this command sets the global maths sizes, as it can only be used in the document preamble.

But it's fairly easy to use: `\DeclareMathSizes{ds}{ts}{ss}{sss}`, where *ds* is the *display size*, *ts* is the *text size*, etc. The values you input are assumed to be point (pt) size.

Note that the changes only take place if the value in the first argument matches the current document text size. It is therefore common to see a set of declarations in the preamble, in the event of the main font being changed. E.g.,

```
\DeclareMathSizes{10}{18}{12}{8}   % For size 10 text
\DeclareMathSizes{11}{19}{13}{9}   % For size 11 text
\DeclareMathSizes{12}{20}{14}{10}  % For size 12 text
```

24 requires the `amsmath` package

25 Chapter 27.6.1 on page 316

28.10. Forcing \displaystyle for all math in a document

Put

```
\everymath{\displaystyle}
```

before

```
\begin{document}
```

to force all math to

```
\displaystyle
```

.

28.11. Adjusting vertical white space around displayed math

There are four parameters that control the vertical white space around displayed math:

```
\abovedisplayskip=12pt
\belowdisplayskip=12pt
\abovedisplayshortskip=0pt
\belowdisplayshortskip=7pt
```

Short skips are used if the preceding line ends, horizontally, before the formula. These parameters must be set after

```
\begin{document}
```

.

28.12. Notes