# 15 Math Extensions with AMS-LaTeX

The American Mathematical Society, AMS, has supported the development and usage of TeX since its first release. Shortly after TeX 82 became available, the AMS produced a macro package for generating a special format `amstex`, described in *The Joy of TeX* by Spivak (1990). The macros in this AMS-TeX complement the mathematical typesetting features of Plain TeX by adding additional ones and simplifying others.

However, AMS-TeX is not a documentation preparation system like LaTeX, describing the logical layout of a document by means of markup commands, but rather is simply an extension to Plain TeX.

The great popularity of TeX as a text formatting program is due primarily to the availability of LaTeX as a user-friendly interface to the underlying TeX machinery. Many authors therefore asked the AMS to provide LaTeX with the same mathematical features as in AMS-TeX. The AMS-LaTeX project was thus launched in 1987, with version 1.0 completed three years later by Frank Mittelbach and Rainer Schöpf, together with Michael Downes of the AMS. AMS-LaTeX has been fully converted to LaTeX $2_\varepsilon$ with version 1.2 in 1995.

The AMS-LaTeX collection consists of three parts: packages for extending mathematical typesetting; extra classes for articles and books published by the AMS; and supplemental fonts for additional symbols, math alphabets, and Cyrillic fonts.

In the next sections, we give an overview of the math extensions available in the package `amsmath`; a more detailed user's manual is delivered with the collection, in the file `amsldoc.tex`. How to invoke the additional fonts is described below in Section 15.4 or in the AMS-supplied manual found in the file `amsfndoc.tex`. The extra classes will not be described in this book; AMS authors should refer to the instructions in the document `instr-l.tex` provided with AMS-LaTeX.

## 15.1 Invoking AMS-LaTeX

**Package:** `amsmath`

If the `\documentclass` statement at the beginning of the LaTeX document selects one of the AMS classes `amsbook`, `amsart`, or `amsproc`, then most of the AMS-LaTeX features are loaded automatically. These features may still be employed with other classes by including the main extension package `amsmath` by means of

`\usepackage[`*options*`]{amsmath}`

in the document's preamble. The list of allowable options is described below in Section 15.2.8.

The `amsmath` package defines many of the new math typesetting features itself, but it also loads a number of other packages from the $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX collection that contain further extensions, such as `amsopn`, `amstext`, and `amsbsy`. These packages could be loaded separately without `amsmath` if only their limited features are wanted. On the other hand, the packages `amscd` and `amsthm` are not included in `amsmath` and must be loaded explicitly if their features are desired. Simply add their names to the list of packages in `\usepackage`.

The following section describes the new commands and environments made available with `amsmath` and its associated packages. We call these the standard features of $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX. Further extensions added by other packages are explained afterwards.

**!** The examples in this chapter employ the user-defined commands `\mi`, `\me`, and `\dif`, defined in Section 7.4.10, for printing upright i, e, and d in math mode.

# 15.2 Standard features of $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX

This relatively long section presents those standard features of $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX that are activated by loading the `amsmath` package or by selecting one of the $\mathcal{A}_\mathcal{M}\mathcal{S}$ classes `amsart`, `amsbook`, or `amsproc`.

## 15.2.1 Additional font switching commands

**Package:** `amsbsy`

Standard LaTeX provides the math alphabet commands `\mathcal`, `\mathrm`, `\mathbf`, `\mathsf`, `\mathit`, `\mathtt` for changing fonts within math mode (Section 7.4.2). With $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX, one may also use the commands

`\boldsymbol{`*symbol*`}` and `\pmb{`*symbol*`}`

to print *symbol* in a boldface, provided there is an appropriate bold font for it. Whereas the command `\mathbf` sets only Latin letters, numbers, and Greek uppercase letters in bold, these commands also affect math symbols and Greek lowercase letters. Compare the result of `$\mathbf{\nabla\times V\,d\sigma}$` ($\mathbf{\nabla\times V\,d\sigma}$) with that of `$\boldsymbol{\nabla\times V\,d\sigma}$` ($\boldsymbol{\nabla\times V\,d\sigma}$). With the standard LaTeX command `\mathbf`, only the letters *V* and *d* appear bold, while the other characters remain in normal weight. Not only that, these letters are upright, not italic as required by international standards. With `\boldsymbol`, all symbols are bold and italic.

Those symbols for which no boldface font exists will remain in normal weight with the `\boldsymbol` command. Such symbols are, for example, those that come in two sizes (Section 7.3.7) like $\sum$, $\int$, $\bigcup$, and so on. The command `\pmb` (poor man's bold) simulates a boldface even for these symbols by printing them several times slightly displaced. The result of `$\pmb{\sum\;\int\;\bigcup}$` is $\pmb{\sum\;\int\;\bigcup}$.

These commands are defined in the package `amsbsy`, which may be loaded separately without `amsmath`.

**Package:** `amstext`

A short piece of normal text can be given within a formula with

`\text{`*short_text*`}`

In contrast to `\mbox{`*short_text*`}` for standard LaTeX, the `\text` command switches font size when used as superscript or subscript. Thus `.._{\text{Word}}` sets $_{\text{Word}}$ lower and changes to `\scriptstyle` font size. To achieve the same effect in standard LaTeX, the font size must be given explicitly, as `.._{\mbox{\scriptstyle Word}}`. Furthermore, the name `\text` is more precise than `\mbox`.

This command is defined in the package `amstext`. Like `amsbsy`, it may be loaded on its own without `amsmath` if none of the other $\mathcal{AMS}$-LaTeX features are wanted.

Another command to insert normal text inside a displayed equation is

`\intertext{`*insert_text*`}`

The *insert_text* is inserted as a left-justified line of text between those of the formula. The alignment of the formula lines remains unaffected, something that is not guaranteed if one closes the displayed equation, inserts text, and reopens the equation. For example:

$$\begin{aligned}(x+\mathrm{i}y)(x-\mathrm{i}y) &= x^2 - \mathrm{i}xy + \mathrm{i}xy - \mathrm{i}^2y^2\\ &= y^2 + y^2\quad\text{since}\quad \mathrm{i}^2=-1\quad\text{is true.}\end{aligned}$$

On the other hand

$$\begin{aligned}(x+\mathrm{i}y)^2 &= x^2 + 2\mathrm{i}xy + \mathrm{i}^2y^2 = x^2 + 2\mathrm{i}xy - y^2\\ (x-\mathrm{i}y)^2 &= x^2 - 2\mathrm{i}xy + \mathrm{i}^2y^2 = x^2 - 2\mathrm{i}xy - y^2\end{aligned}$$

```
\begin{align*}
 (x+\mi y)(x-\mi y) & = x^2 - \mi xy + \mi xy - \mi ^2y^2\\
            & = y^2 + y^2\quad\text{since}\quad \mi ^2=-1
            \quad\text{is true.}\\
 \intertext{On the other hand}
 (x +\mi y)^2 & = x^2 + 2\mi xy +\mi ^2y^2 = x^2 + 2\mi xy-y^2\\
 (x -\mi y)^2 & = x^2 - 2\mi xy +\mi ^2y^2 = x^2 - 2\mi xy-y^2
\end{align*}
```

The alignment of the lines on the first equals sign is maintained through the interruption with the line of text 'On the other hand'. The environment `align` (Section 15.2.6) is one of several new ones provided by $\mathcal{AMS}$-LaTeX to replace the standard LaTeX environment `eqnarray`. Note that `\intertext` may only be issued immediately after the `\\` for starting a new line.

### 15.2.2 Multiple mathematical symbols

Mathematical formulas often require the same symbol to appear several times, such as multiple integral signs, or symbols and arrows to be stacked above or below a mathematical expression. Usually the distances between these symbols depends on the mathematical meaning, something that LaTeX cannot automatically recognize. AMS-LaTeX provides a number of structures to help the user find the right spacing without tedious trial and error.

#### Multiple integrals

With AMS-LaTeX, the commands `\iint`, `\iiint`, `\iiiint`, and `\idotsint` output multiple integrals, with upper and lower limits being added in the usual manner. In text formulas, they are printed as $\iint$, $\iiint$, $\iiiint$, $\int\cdots\int$, while in displayed formulas, they appear as

$$\iint\limits_S f(x,y)\,\mathrm{d}S$$ `\[ \iint\limits_S f(x,y)\,\dif S \]`

$$\iiint\limits_V f(x,y,z)\,\mathrm{d}V$$ `\[ \iiint\limits_V f(x,y,z)\,\dif V\]`

$$\iiiint\limits_G f(x,y,z,t)\,\mathrm{d}G$$
```
\[ \iiiint\limits_G f(x,y,z,t)
                  \,\dif G \]
```

$$\idotsint\limits_U f(x_1,\ldots,x_k)\,\mathrm{d}U$$
```
\[ \idotsint\limits_U
   f(x_1,\ldots,x_k)\,\dif U \]
```

whereby the command `\limits` may be left off if the option `intlimits` has been specified when `amsmath` was loaded (see Section 15.2.8).

#### Multiline limits

One often needs multiline limits or indices for summations, as in the examples below.

$$\Delta_{\substack{p_1p_2\cdots p_{n-k}\\q_1q_2\cdots q_{n-k}}} \qquad \sum_{\substack{k_0,k_1,\ldots\geq 0\\k_0+k_1+\cdots=0}} a_{0k_0}a_{1k_1}\cdots$$

AMS-LaTeX provides a command `\substack` for this purpose, with syntax

`\substack{`*1st line*`\\`*2nd line*`\\...\\` *last line*`}`

where the command must immediately follow the `^` or `_` shifting commands, and be entirely enclosed in curly braces `{ }`. The index for the left-hand example above was generated with

```
\Delta_{\substack{p_1p_2\cdots p_{n-k}\\q_1q_2\cdots q_{n-k}}}
```

The lines of text printed by `\substack` are centered horizontally, as is apparent in the right-hand example above. This was produced with

```
\[ \sum_{\substack{k_0,k_1,\ldots\ge0\\ k_0+\k_1+
      \cdots=0}}a_{0k_0} a_{1k_1}\cdots \]
```

On the other hand, the environment `subarray` offers more control over the horizontal alignment:

`\begin{subarray}{`*pos*`}` *1st line*`\\`*2nd line*`\\...\\` *last line*
`\end{subarray}`

The argument *pos* may be `c` for centered or `l` for left-justified lines.

$$\sum_{\begin{subarray}{l} i\in\Lambda\\ i<j<n \end{subarray}} P(i,j)$$

```
\[ \sum_{\begin{subarray}{l} i\in\Lambda\\
      i<j<n \end{subarray}} P(i,j) \]
```

### Special limits

To add a differential prime sign to a summation symbol without a lower limit, such as $\sum' E_n$, in a displayed formula, one can easily give

```
\[ \sum\nolimits' E_n \]
```

$$\sum\nolimits' E_n$$

However, if there is to be an upper limit as well, the prime can only be added with much fiddling. The `\sideset` command simplifies this task considerably.

$$\sideset{}{'}\sum_{n=0}^\infty (2n+1) E_{2n+1}$$

```
\[ \sideset{}{'}\sum_{n=0}^\infty
   (2n+1) E_{2n+1} \]
```

The complete syntax for this command is

`\sideset{`*pre*`}{`*post*`}\`*symbol*

where *pre* and *post* are the superscripts and subscript commands to be added before and after the *symbol*, respectively. They must contain the raising and lowering operators `^` and `_`.

The product symbol $\prod$ is given daggers above and asterisks below with `\sideset{_\dag^*}{_\dag^*}\prod`, as shown at the right. $\sideset{_\dag^*}{_\dag^*}\prod$

Additional shifting commands are

`\overset{`*char*`}{\`*symbol*`}` and `\underset{`*char*`}{\`*symbol*`}`

which place the arbitrary *char* above or below the *symbol* in the size appropriate for superscripts and subscripts. Thus `$\overset{*}{X}$` produces $\overset{*}{X}$ and `$\underset{*}{X}$` yields $\underset{*}{X}$.

### Extended arrows

The `amsmath` package provides a number of commands to produce extra long arrows for combining with mathematical expressions. The commands

\overleftarrow{*expr*} \underleftarrow{*expr*}
\overrightarrow{*expr*} \underrightarrow{*expr*}
\overleftrightarrow{*expr*} \underleftrightarrow{*expr*}

produce lengthened arrows pointing left and right, as well as double arrows above and below the mathematical expression *expr*.

$$\begin{aligned} \overrightarrow{ABCD} &= \underrightarrow{AB} + \underrightarrow{BC} + \underrightarrow{CD} \\ \overleftarrow{ABCD} &= \underleftarrow{DC} + \underleftarrow{CB} + \underleftarrow{BA} \\ \overleftrightarrow{ABCD} &= \underleftrightarrow{DCAB} \end{aligned}$$

```
\begin{eqnarray*}
  \overrightarrow{ABCD} & = &
    \underrightarrow{AB} +
    \underrightarrow{BC} +
    \underrightarrow{CD} \\
  \overleftarrow{ABCD} & = &
    \underleftarrow{DC} ...
\end{eqnarray*}
```

The lower arrows are actually too close to the expression, as is seen on the right-hand side of the last example. This should also be the case for all the other examples with a lower arrow, except that we have added a strut to push them down somewhat:

`\underrightarrow{\rule[-2pt]{0pt}{2pt}AB} ...` and
`\underleftarrow{\rule[-2pt]{0pt}{2pt}DC} ...`

A smaller font size will be used for the arrows when they appear in exponents, indices, superscripts, and subscripts:

$$\int_{\overrightarrow{0.2\pi}} r\,d\varphi = 2\pi r$$

```
\int_{\overrightarrow{0.2\pi}}
    r\,d\varphi = 2\pi r
```

There are two more commands for horizontal arrows of variable length:

\xleftarrow[*below*]{*above*} \xrightarrow[*below*]{*above*}

which place the mandatory *above* in superscript size over the arrow and the optional *below* in subscript size beneath it.

$$A \xleftarrow{n+\mu-1} B \xrightarrow[T]{n\pm i-1} C$$

```
\[ A \xleftarrow{n+\mu-1} B
     \xrightarrow[T]{n\pm i-1} C \]
```

The package `amscd` (Section 15.3.2) offers further possibilities for combining arrows and text.

### Multiple dots over symbols

Three or four dots in a row over a symbol are often used for time derivatives of third and fourth order. They can be placed with the commands

\dddot{*sym*} and \ddddot{*sym*}

In this way $\dddot{u}$ and $\ddddot{u}$ are produced with `$\dddot{u}$` and `$\ddddot{u}$`.

### Continuation dots

In standard LaTeX, one has the commands `\ldots` and `\cdots` for printing three continuation dots, either on the baseline or raised to the center of the line. AMS-LaTeX offers a number of additional possibilities. The most general of these is the `\dots` command that adjusts the vertical height according to the symbol that follows it. If this is an equals sign or binary operator, such as + or −, the dots are raised, as with `\cdots`, otherwise they are on the baseline, as with `\ldots`.

`$a_0+a_1+\dots+a_n$` ⟹ $a_0 + a_1 + \cdots + a_n$
`$a_0,a_1,\dots,a_n$` ⟹ $a_0, a_1, \ldots, a_n$

If the continuation dots come at the end of a formula, there is no following symbol to determine the height of the dots. In this case, one must manually indicate the height by means of one of the commands `\dotsc` (comma), `\dotsb` (binary), `\dotsm` (multiplication), or `\dotsi` (integral).

The `\dotsc` command is to be used with commas, so `$A_1,A_2,\dotsc$` produces $A_1, A_2, \ldots$; the `\dotsb` command sets them for a binary operator, thus `$A_1+A_2+\dotsb$` yields $A_1 + A_2 + \cdots$; the command `\dotsm` is actually identical to `\dotsb`, but it is logically meant to be applied to multiplication, `$A_1A_2\dotsm$` makes $A_1A_2\cdots$; finally `\dotsi` places the dots at the mean height of an adjacent integral sign.

`\[ \int_{A_1}\int_{A_2}\dotsi \]` $$\int_{A_1}\int_{A_2}\cdots$$

## 15.2.3 Fractions

The TeX fraction commands `\atop`, `\choose`, and others may be allowed in standard LaTeX, but not in AMS-LaTeX. With the `amsmath` package, only those fraction commands described here may be used.

### Basic fraction commands

In addition to the regular LaTeX command `\frac{`*over*`}{`*under*`}`, AMS-LaTeX provides the commands `\tfrac` and `\dfrac` with the same syntax. These are effectively the same as `\frac` but with the font set to `\textstyle` or `\displaystyle`, respectively (Section 7.5.2). We demonstrate their effects with some examples from the AMS-LaTeX user's manual.

```
\[ \frac{1}{k}\log_2 c(f)\qquad
   \tfrac{1}{k}\log_2 c(f)\qquad
   \sqrt{\frac{1}{k}\log_2 c(f)}\qquad
   \sqrt{\dfrac{1}{k}\log_2 c(f)}   \]
```

$$\frac{1}{k}\log_2 c(f)\qquad \tfrac{1}{k}\log_2 c(f)\qquad \sqrt{\frac{1}{k}\log_2 c(f)}\qquad \sqrt{\dfrac{1}{k}\log_2 c(f)}$$

### Binomial expressions

A binomial expression looks something like a fraction but is enclosed in round parentheses and is missing the horizontal rule. The basic command in the `amsmath` package is

`\binom{`*over*`}{`*under*`}`

which functions in the same way as `\frac` and the other fraction commands.

```
\[ \binom{n+1}{k} = \binom{n}{k}
   + \binom{n}{k-1 \]
```

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

Similarly, the commands `\tbinom` and `\dbinom` are analogous to `\tfrac` and `\dfrac`.

### User-defined fractions

The `amsmath` package provides a powerful tool for defining fraction-like structures:

`\genfrac{`*left_brk*`}{`*right_brk*`}{`*thickness*`}{`*mathsize*`}{`*over*`}{`*under*`}`

where *left_brk* and *right_brk* are the parenthesis characters on the left and right, *thickness* is the thickness of the horizontal line, and {*mathsize*} is a number 0–3 representing the math sizes `\displaystyle`, `\textstyle`, `\scriptstyle`, and `\scriptscriptstyle`, respectively. The last two arguments, *over* and *under*, are the texts in the two parts of the fraction, the same arguments as in `\frac` and `\binom`.

If the *thickness* is left blank, the standard thickness for LaTeX fractions is used. If *mathsize* is empty, the font size is determined automatically by the normal rules in Section 7.5.2.

Rather than repeating `\genfrac` with the same first four arguments time and again, one should define new fraction commands with it. For example, the following definitions are given in `amsmath.sty`:

```
\newcommand{\frac}[2]{\genfrac{}{}{}{}{#1}{#2}}
\newcommand{\dfrac}[2]{\genfrac{}{}{}{0}{#1}{#2}}
\newcommand{\tfrac}[2]{\genfrac{}{}{}{1}{#1}{#2}}
\newcommand{\binom}[2]{\genfrac{(}{)}{0pt}{}{#1}{#2}
```

As a further example, consider the redefinition of the command `\frac`

```
\renewcommand{\frac}[3][]{\genfrac{}{}{#1}{}{#2}{#3}}
```

in which the line thickness is now an optional first argument; without this optional argument, the command behaves as normal. Thus

```
\[ \binom{n}{m} =
   \frac[2pt]{n!}{M!(n-m)!} \]
```

yields

$$\binom{n}{m} = \genfrac{}{}{2pt}{}{n!}{M!(n-m)!}$$

### Continued fractions

Continued fractions can be made in AMS-LaTeX with the command

\cfrac[*pos*]{*over*}{*under*}

whereby the denominator *under* may contain further \cfrac commands.

```
\[ a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 +
         \cfrac{1}{a_3 + \cfrac{1}{a_4 +
         \dotsb }}}}  \]
```

produces

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

If the optional argument *pos* is missing, the numerator *over* is centered on the horizontal rule; otherwise it may take values of l or r to left or right justify the numerator.

## 15.2.4 Matrices

Standard LaTeX possesses the environment array for producing arrays and matrices. The amsmath package provides the additional environments pmatrix, bmatrix, Bmatrix, vmatrix, and Vmatrix, which automatically add the enclosing braces (), [], {}, | |, and ‖ ‖ around the array, and in the right size. For completeness, there is also a matrix environment with no braces.

In contrast to the standard array environment (Section 6.2.1), these matrix environments do not require an explicit column specification as argument. By default, up to 10 centered columns may be used without any argument. (This maximum number may be changed by giving the special counter MaxMatrixCols a new value with either \setcounter or \addtocounter.) Otherwise the matrix environments are used in the same way as the array environment.

The following example is taken from Section 7.4.3 on page 136 and is recast here using AMS-LaTeX constructs.

$$\sum_{p_1<p_2<\dots<p_{n-k}}^{(1,2,\dots,n)} \Delta_{\substack{p_1p_2\dots p_{n-k}\\p_1p_2\dots p_{n-k}}} \sum_{q_1<q_2<\dots<q_k} \begin{vmatrix} a_{q_1q_1} & a_{q_1q_2} & \dots & a_{q_1q_k}\\ a_{q_2q_1} & a_{q_2q_2} & \dots & a_{q_2q_k}\\ \hdotsfor{4}\\ a_{q_kq_1} & a_{q_kq_2} & \dots & a_{q_kq_k} \end{vmatrix}$$

```
\[ \sum_{p_1<p_2<\dots<p_{n-k}}^{(1,2,\dots,n)}
   \Delta_{\substack{p_1p_2\dots p_{n-k}\\p_1p_2
   \dots p_{n-k}}}
```

```
\sum_{q_1<q_2<\dots<q_k}
\begin{vmatrix}
  a_{q_1q_1} & a_{q_1q_2} & \dots & a_{q_1q_k}\\
  a_{q_2q_1} & a_{q_2q_2} & \dots & a_{q_2q_k}\\
  \hdotsfor[2.0]{4}\\
  a_{q_kq_1} & a_{q_kq_2} & \dots & a_{q_kq_k}
\end{vmatrix}             \]
```

Comparing this input text with that on page 136, one sees that it is simpler and easier to follow. The only new command used here is `\hdotsfor`, which has the syntax

`\hdotsfor[`*stretch*`]{`*n*`}`

and which prints a continuous line of dots through *n* columns. The optional argument *stretch* is a multiplicative number to increase the dot density, being 1.0 by default.

```
\[ \begin{matrix} a & b & c & d & e\\
                  x & \hdotsfor{3} & z
   \end{matrix}  \]
```

$$\begin{matrix} a & b & c & d & e\\ x & \hdotsfor{3} & z \end{matrix}$$

Compare the standard dot spacing above with that from `\hdotsfor[2.0]` in the previous example.

The initial letter of each of the *x*`matrix` environments indicates the type of braces that enclose it: `pmatrix` for (round) parentheses, `bmatrix` for (square) brackets, `Bmatrix` for (curly) braces, `vmatrix` for vertical lines, and `Vmatrix` for double vertical lines. They appear as

$$\begin{matrix} r & s & t\\ u & v & w\\ x & y & z \end{matrix} \quad \begin{pmatrix} r & s & t\\ u & v & w\\ x & y & z \end{pmatrix} \quad \begin{bmatrix} r & s & t\\ u & v & w\\ x & y & z \end{bmatrix}$$

$$\begin{Bmatrix} r & s & t\\ u & v & w\\ x & y & z \end{Bmatrix} \quad \begin{vmatrix} r & s & t\\ u & v & w\\ x & y & z \end{vmatrix} \quad \begin{Vmatrix} r & s & t\\ u & v & w\\ x & y & z \end{Vmatrix}$$

where each matrix has been produced with

```
\[ \begin{xmatrix} r & s & t\\ u & v & w\\ x & y & z
   \end{xmatrix}  \]
```

where *x*`matrix` is set to `matrix`, `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, and `Vmatrix` one after the other.

To generate a small array within a text formula, one can apply the `smallmatrix` environment. In this way $\bigl( \begin{smallmatrix} a & b & c\\ e & m & r \end{smallmatrix} \bigr)$ can be made with

```
$ \bigl( \begin{smallmatrix} a & b & c\\ e & m & r
        \end{smallmatrix} \bigr) $
```

### 15.2.5 User extensions and fine adjustments

#### Function names

**Package:** `amsopn`

Standard LaTeX recognizes a number of predefined function names (Section 7.3.8) that are printed in math made by placing a backslash in front of that name: arccos, arcsin, arctan, arg, cos, cosh, cot, coth, csc, deg, det, dim, exp, gcd, hom, inf, ker, lg, lim, liminf, limsup, ln, log, max, min, Pr, sec, sin, sinh, sup, tan, tanh. Not only do these names appear in an upright font, as is required for function names, but the spacing with adjacent parts of the mathematical expression is adjusted automatically.

AMS-LaTeX provides some more function names, as variations on the standard `\lim` name:

| | | | |
|---|---|---|---|
| `\varlimsup` | $\varlimsup$ | `\varinjlim` | $\varinjlim$ |
| `\varliminf` | $\varliminf$ | `\varprojlim` | $\varprojlim$ |

These functions may take on limits with the raising and lowering operators ^ and _; for example, `\varliminf_{n\to\infty}` for $\varliminf_{n\to\infty}$.

It is also possible to define new function names with the same font and spacing properties as the predefined ones. The command

`\DeclareMathOperator{\`*cmd*`}{`*name*`}`

which may only be issued in the preamble, before `\begin{document}`, defines a command `\`*cmd* that prints the function name *name*. For example, to define a function name `\doit`, give

```
\DeclareMathOperator{\doit}{doit}
```

and then `$A=3\doit^2(B)$` yields $A = 3\operatorname{doit}^2(B)$. Note that superscripts and subscripts are printed beside the operator; if they are to be printed as limits, that is, above and below the operator in displayed math mode, use the *-form to define them. For example:

```
\DeclareMathOperator*{\Lim}{lim}
\[ \Lim_{n\to-\infty}^{n\to+\infty}\]
```

$$\lim_{n\to-\infty}^{n\to+\infty}$$

The *name* text need not be identical to the command name. In particular, it may contain special characters not allowed in command names.

Modulo expressions are printed in standard LaTeX with `\bmod` and `\pmod` commands, and are complemented in AMS-LaTeX by `\mod` and `\pod`. The possibilities are:

| | |
|---|---|
| $z \equiv x+y \bmod{n^2}$ | `z \equiv x+y \bmod{n^2}` |
| $z \equiv x+y \pmod{n^2}$ | `z \equiv x+y \pmod{n^2}` |
| $z \equiv x+y \mod{n^2}$ | `z \equiv x+y \mod{n^2}` |
| $z \equiv x+y \pod{n^2}$ | `z \equiv x+y \pod{n^2}` |

The automatic parentheses are missing with `\mod`, while with `\pod` the name 'mod' is omitted. Furthermore, `\pmod` is redefined for text formulas to reduce the preceding space: `$y\pmod{a+b}$`: $y \pmod{a+b}$.

The `\DeclareMathOperator` command and additional function names are defined in the `amsopn` package, which may be loaded on its own without `amsmath`.

### Fine-tuning roots

The positioning of an index to a root sign is not always ideal under standard LaTeX. In $\sqrt[\beta]{k}$, for example, the $\beta$ could be somewhat higher and shifted slightly to the right. The $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-LaTeX commands

`\leftroot{`*shift*`}`   `\uproot{`*shift*`}`

cause such manual displacements, where *shift* is a number specifying the size in small, internal units. Negative numbers represent a shift in the opposite direction. Compare the above standard result with that of

`$\sqrt[\leftroot{-1}\uproot{3}\beta]{k}$`   $\sqrt[\leftroot{-1}\uproot{3}\beta]{k}$

The size of the root sign depends on its contents. If they hang below the baseline, the root sign extends lower down than for contents that have no depth. Note the differences between $\sqrt{x}$, $\sqrt{y}$, and $\sqrt{z}$. Some publishers want all root signs to be at the same height, as $\sqrt{x} + \sqrt{\smash[b]{y}} + \sqrt{z}$. This is accomplished with the TeX command `\smash`, which places its argument in a box of zero height and depth. The $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-LaTeX version of this command allows an optional argument `b` or `t` to zero only the depth or height, respectively. The above example is produced with `$\sqrt{\smash[b]{y}}$`. The option `b` is taken because we want only the depth to be ignored, not the height of the letter $y$.

### Spacing adjustment

With standard LaTeX, there are a number of commands to fine-tune the spacing in a math formula (Section 7.5.1). These are `\,`, `\:`, `\;`, `\quad`, and `\qquad` for increasing amounts of positive spacing, and `\!` for negative spacing. With the `amsmath` package, the first three still exist, but may also be called with the more obvious names `\thinspace`, `\medspace`, and `\thickspace`. There is also `\negthinspace` as an alias for `\!`.

The complete set of spacing commands are summarized in the table below taken from the $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-LaTeX manual.

| Short form | Command name | Demo | Short form | Command name | Demo |
|---|---|---|---|---|---|
| `\,` | `\thinspace` | ⌟\,⌞ | `\!` | `\negthinspace` | ⌟\!⌞ |
| `\:` | `\medspace` | ⌟\:⌞ | | `\negmedspace` | ⌟⌞ |
| `\;` | `\thickspace` | ⌟\;⌞ | | `\negthickspace` | ⌟⌞ |
| | `\quad` | ⌟ ⌞ | | | |
| | `\qquad` | ⌟   ⌞ | | | |

The general math spacing command is

```
\mspace{mu}
```

which inserts space in mathematical spacing units 'mu' (=1/18 em). For example, `\mspace{-9mu}` puts in negative spacing of 1/2 em.

#### Vertical bars

In standard LaTeX, the commands `|` and `\|` are used for single and double vertical bars, | and ‖. However, these symbols are often used as delimiters (that is, like braces), in which case different spacing requirements are needed. In particular, a distinction must be made between the left and right delimiter in expressions like $|a|$ and $\|v\|$. The LaTeX commands are only appropriate for single appearances, like $p|q$ or $f(t,x)|_{t=0}$.

The `amsmath` package defines the delimiter commands `\lvert`, `\rvert` for a single bar, and `\lVert`, `\rVert` for a double bar. They are useful for defining commands that set their arguments in such delimiters, as

```
\newcommand{\abs}[1]{\lvert#1\rvert}
\newcommand{\norm}[1]{\lVert#1\rVert}
```

Now `$\abs{a}$` produces $\lvert a\rvert$ and `$\norm{v}$` $\lVert v\rVert$.

A similar recommendation can be made for the standard commands `\langle` and `\rangle`. By defining

```
\newcommand{\mean}[1]{\langle#1\rangle}
```

one gives `$\mean{x}$` to generate $\langle x\rangle$, rather than `$<x>$` which produces $<x>$.

#### Boxed formulas

A formula may be placed in a box with the command

```
\boxed{formula}
```

For example,

```
\[ \boxed{\int_0^\infty f(x)\,\dif x \approx
     \sum_{i=1}^n w_i \me^{x_i} f(x_i)}  \]
```

produces

$$\boxed{\int_0^\infty f(x)\,\mathrm{d}x \approx \sum_{i=1}^n w_i \mathrm{e}^{x_i} f(x_i)}$$

### 15.2.6 Multiline equations

Equations consisting of several lines that are horizontally aligned at set points, such as the equals sign, can be generated in standard LaTeX with the `eqnarray` and `eqnarray*` environments (Section 7.4.7). Many authors consider these to be far too limited for publications with complicated multiline equations. AMS-LaTeX therefore provides a range of further alignment environments for formulas extending over a single line:

`align   gather   falign   multline   alignat   split`

With the exception of `split`, all exist in a standard and a *-form. As for `eqnarray`, the standard form adds an automatic equation number to each line, while the *-form does not. The standard LaTeX `equation` environment for single-line formulas is also available in AMS-LaTeX in a *-form. It may be used in combination with multiline environments.

### Common features of alignment environments

All the alignment, or multiline, environments switch to math mode at the start and back to text mode at the end, except for `split`, which must be called in math mode. A new line is forced in the formula with the \\ command, as usual; an optional argument \\[*len*] can be added to increase the line spacing by *len*, again as usual.

The automatic numbering with the standard forms can be suppressed for single lines by adding `\notag` before the \\ line break. Alternatively, the line can be given a desired marker with `\tag{`*mark*`}`. For example, with `\tag{$\dag$}`, the marker is (†). Using the *-form instead, the marker text is printed without the parentheses.

The vertical position of the equation number or marker is shifted automatically if there is not enough room on the line for it. This shift can be manually adjusted with the command

`\raisetag{`*len*`}`

which moves the marker upwards by *len* for that line only. A negative value moves it downwards.

### The `multline` environment

The `multline` environment is a variant of the `equation` environment for *single* formulas that are too long for one line. The line breaks occur where the user forces them with the \\ command. The first line is left justified, the last right justified, and lines in between are centered. However, if the option `fleqn` has been given, all the lines appear left justified.

The equation number, if present, appears at the right of the *last* line by default or if the option `reqno` has been selected; if the option `leqno` has been chosen, the number is placed at the left of the *first* line. (See Section 15.2.8 for the `amsmath` options.)

The AMS classes (not dealt with here) put equation numbers at the left by default, something that leads to some confusion.

It is possible to shift individual lines fully to the left or right with the commands `\shoveleft{`*formula*`}` and `\shoveright{`*formula*`}`. The entire formula text for that line, except the terminating \\, is placed in their arguments.

The left and right margins for the formula are set by the length parameter `\multlinegap`, which is initially 10 pt. This may be altered by the user with the `\setlength` or `\addtolength` commands.

An equation with five lines could be broken to look as follows:

First line — left justified

Second line — horizontally centered

Third line — pushed to the left

Fourth line — pushed to the right

Last line — right justified (15.1)

```
\begin{multline}
  \framebox[.75\columnwidth]{First line --- left justified}\\
  \framebox[.6\columnwidth]{Second line --- horizontally
       centered}\\
  \shoveleft{\framebox[.6\columnwidth]{Third line --- pushed
       to the left}}\\
  \shoveright{\framebox[.6\columnwidth]{Fourth line --- pushed
       to the right}}\\
  \framebox[.75\columnwidth]{Last line --- right justified}
\end{multline}
```

A real equation would contain mathematical expressions and not the `\framebox` commands in the above demonstration.

### The `split` environment

Like `multline`, the `split` environment is meant for a single equation that does not fit on one line. Line breaks are again forced with the \\ command; the difference is that in each line there is an alignment marker & such that the lines are horizontally positioned to line up the markers.

The `split` environment does not switch into math mode, nor does it produce an equation number. It is intended to be applied within another math environment, such as `equation` or `gather`. This is why there is an `equation*` environment in $\mathcal{AMS}$-LaTeX.

The equation number, if present, is provided by the outer environment. It is applied to the entire multiline formula, which by default, or with the option `centertags`, is centered on the group of lines. With the option `tbtags`, it is placed either at the left of the first line, or at the right of the last line, depending on the further options `leqno` and `reqno`, respectively. (See Section 15.2.8.)

$$\begin{split} H_c={}&\frac{1}{2n}\sum_{l=0}^n (-1)^l (k-l)^{p-2} \sum_{l_1+\dots+l_p=l} \prod_{i=1}^p \binom{n_i}{l_i}\\ &\times[(k-l) - (k_i-l_i)]^{k_i-l_i}\times \Bigl[(k-l)^2 - \sum_{j=1}^p (k_i-l_i)^2\Bigr] \end{split} \tag{15.2}$$

```
\begin{equation}\begin{split}
  H_c={}&\frac{1}{2n}\sum_{l=0}^n (-1)^l (k-l)^{p-2}
       \sum_{l_1+\dots+l_p=l} \prod_{i=1}^p \binom{n_i}{l_i}\\
     &\times[(k-l) - (k_i-l_i)]^{k_i-l_i}\times
       \Bigl[(k-l)^2 - \sum_{j=1}^p (k_i-l_i)^2\Bigr]
\end{split}\end{equation}
```

The alignment has been chosen to be just after the equals sign. In order not to interfere with the normal spacing around equals signs, a dummy {} has been inserted afterwards, before the &. If the equation consists of several lines all beginning with =, then it would be better to put the alignment character before the = so that all equals signs align. Note the centered equation number at the right.

### The gather environment

The `gather` environment switches to math mode, centering each of its formula lines without any alignment. The formula lines are separated by \\ commands. Each line receives an equation number, unless the *-form has been used or `\notag` has been issued in that line.

$$\frac{1}{2} + \left(\frac{2}{3}\right)^4 + \left(\frac{3}{4}\right)^9 + \dots + \left(\frac{n}{n+1}\right)^{n^2} + \dotsb = \sum_{n=1}^\infty \left(\frac{n}{n+1}\right)^{n^2} \tag{15.3}$$

$$\text{converges since}\quad\lim_{n\to\infty} \sqrt[n]{\left(\frac{n}{n+1}\right)^{n^2}} = \lim_{n\to\infty} \left(\frac{1}{1+\frac{1}{n}}\right)^n = \frac{1}{\mathrm{e}} < 1 \quad\text{root condition}$$

$$2 + \frac{3}{4} + \frac{4}{9} + \dots + \frac{n+1}{n^2} + \dotsb = \sum_{n=1}^\infty \frac{n+1}{n^2} \tag{15.4}$$

$$\text{diverges since}\quad \int_c^\infty \frac{x+1}{x^2}\,\mathrm{d}x = \left[\ln x - \frac{1}{x}\right]_c^\infty = \infty \qquad \text{(integral condition)}$$

```
\begin{gather}
  \frac{1}{2} + \left(\frac{2}{3}\right)^4 + \left(\frac{3}{4}
  \right)^9 + \dots + \left(\frac{n}{n+1}\right)^{n^2} + \dotsb
  = \sum_{n=1}^\infty \left(\frac{n}{n+1}\right)^{n^2} \\
  \text{converges since}\quad\lim_{n\to\infty}
```

```
  \sqrt[n]{\left(\frac{n}{n+1}\right)^{n^2}} = \lim_{n\to\infty}
  \left(\frac{1}{1 + \dfrac{1}{n}}\right)^n = \frac{1}{\me} < 1
  \tag*{root condition}\\
  2 + \frac{3}{4} + \frac{4}{9} + \dots + \frac{n+1}{n^2} +
  \dotsb = \sum_{n=1}^\infty \frac{n+1}{n^2}\\
  \text{diverges since}\quad\int_c^\infty \frac{x+1}{x^2}\,
  \dif x = \left[ \ln x -\frac{1}{x}\right]_c^\infty = \infty
  \tag{integral condition}
\end{gather}
```

Note the use of `\tag*` and `\tag` (page 282) to add text as markers to the second and fourth lines.

### The `align` environment

The `align` environment is intended for multiple equations with horizontal alignment, usually on an equals sign or equivalent. New lines are indicated with `\\` as usual. Each line is split into aligned columns such that the first column is right justified against the & character; the second left justified; the third column is right justified against the *third* &; the fourth column left justified again, and so on. This is the same as an `array` environment with column specification `{rl rl rl ...}`.

$$\begin{aligned}(x^n)' &= nx^{n-1} & (\mathrm{e}^x)' &= \mathrm{e}^x & (\sin x)' &= \cos x\\ \left(\frac{1}{x^n}\right)' &= -\frac{n}{x^{n+1}} & (a^x)' &= a^x\ln a & (\cos x)' &= -\sin x\\ \left(\sqrt[n]{x}\right)' &= \frac{1}{n\sqrt[n]{x^n-1}} & (\ln x)' &= \frac{1}{x} & (\tan x)' &= \frac{1}{\cos^2 x}\\ & & (\log_a x)' &= \frac{1}{x\ln a} & (\cot x)' &= -\frac{1}{\sin^2 x}\end{aligned}$$

```
\begin{align*}
  \left(x^n\right)' &= nx^{n-1} & \left(\me^x\right)' &= \me^x &
  (\sin x)'         &= \cos x   \\
  \left(\frac{1}{x^n}\right)' &= -\frac{n}{x^{n+1}} &
  \left(a^x\right)' &= a^x\ln a & (\cos x)'        &= -\sin x\\
  \left(\sqrt[n]{x}\right)'   &= \frac{1}{n\sqrt[n]{x^n -1}} &
  (\ln x)'          &= \frac{1}{x} & (\tan x)'
                              &= \frac{1}{\cos^2 x}\\
  & &   (\log_a x)' &= \frac{1}{x\ln a} & (\cot x)'
                              &= -\frac{1}{\sin^2 x}
\end{align*}
```

The `align*` environment is used here to prevent the lines from being numbered, something that is not appropriate for such a collection of formulas. The alignment within each of the three column pairs is on the equals sign. The input for the last line begins with a double && to produce an empty column pair.

Occasionally a set of formulas is to be aligned on several equals signs in one line, as in the equations below for the volume $V$, inertial moment $I_z$, and mass $M$ of an arbitrary body, in Cartesian, cylindrical, and spherical coordinates. In this case, the second and third parts are separated by a double && so that the left-hand sides of these column pairs are empty: The equals signs are always on an odd-numbered alignment marker.

$$\begin{aligned} V &= \int\limits_V \mathrm{d}v &&= \iiint \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z &&= \iiint \rho\,\mathrm{d}x\,\mathrm{d}\rho\,\mathrm{d}\phi \\ &&&&&= \iiint r^2 \sin\theta\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}\phi \qquad (15.5) \\ I_z &= \int\limits_V \rho^2\,\mathrm{d}v &&= \iiint (x^2+y^2)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z &&= \iiint \rho^3\,\mathrm{d}z\,\mathrm{d}\rho\,\mathrm{d}\phi \\ &&&&&= \iiint r^4 \sin^3\theta\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}\phi \qquad (15.6) \\ M &= \int\limits_V \delta\,\mathrm{d}v &&= \iiint \delta\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z &&= \iiint \delta\rho\,\mathrm{d}z\,\mathrm{d}\rho\,\mathrm{d}\phi \\ &&&&&= \iiint \delta r^2 \sin\theta\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}\phi \qquad (15.7) \end{aligned}$$

```
\begin{align}
V   &= \int\limits_V\dif v      &&= \iiint\dif x\,\dif y\,\dif z
   &&= \iiint \rho\,\dif x\,\dif \rho\,\dif \phi \notag \\
&&&&&= \iiint r^2 \sin\theta\,\dif r\,\dif\theta\,\dif\phi\\
I_z &= \int\limits_V \rho^2\,\dif v &&=\iiint(x^2 + y^2)
                                    \,\dif x\,\dif y\,\dif z
   &&= \iiint \rho^3\,\dif z\,\dif \rho\,\dif \phi \notag \\
&&&&&= \iiint r^4\sin^3\theta\,\dif r\,\dif\theta\,\dif\phi\\
M   &= \int\limits_V \delta\,\dif v &&= \iiint \delta
                                    \,\dif x\,\dif y\,\dif z
   &&= \iiint \delta\rho\,\dif z\,\dif\rho\,\dif\phi\notag \\
&&&&&=\iiint\delta r^2\sin\theta\,\dif r\,\dif\theta\,\dif\phi
\end{align}
```

There are two variations on the `align` environment, `falign` and `alignat`. The first has exactly the same syntax as `align` but it inserts so much spacing between the column pairs that the entire line is filled out. The `alignat` environment is just the opposite: No spacing is inserted automatically between the column pairs. It must take the number of column pairs as a mandatory argument, otherwise the syntax of the contents is the same as that for `align`. The example above with the volume, inertial moment, and mass of a body could just as well have been given with

`\begin{alignat}{3}` *formula_text* `\end{alignat}`

(In fact, this is precisely what was done in order to fit it within the line width of this book.)

If one or more columns are empty, as in this example, it is possible to control their widths precisely in the `alignat` environment by adding explicit spacing between the two & characters in one of the lines. See Section 7.5.1 for spacing in math mode.

In summary, for the `align` environment and its variants, the first, third, fifth, ... & characters are alignment markers, while the second, fourth, sixth, ... are column pair separators.

### Nested alignment environments

We have already pointed out on page 283 how the `split` environment is to be placed inside an `equation` environment. The same is true for the environments `aligned` and `gathered`, which may be used as building blocks within formulas. Their contents and behavior are otherwise the same as their related environments.

Both of these environments take an optional argument *pos*

```
\begin{aligned}[pos]    lines  \end{aligned}
\begin{gathered}[pos]   lines  \end{gathered}
```

which takes values of `t` or `b` to determine the vertical alignment (top or bottom) when they appear beside other elements. When no *pos* is given, they are centered. In this way

$$\begin{aligned} \alpha&=aa\\ \beta&=bbbbb\\ \gamma&=g \end{aligned} \qquad\text{versus}\qquad \begin{aligned}[t] \delta&=dd\\ \eta&=eeeeee\\ \varphi&=f \end{aligned} \qquad\text{versus}\qquad \begin{gathered}[b] s= x+y\\ d= u - v - w\\ p = x\circ y \end{gathered}$$

is produced with

```
\begin{equation*}
  \begin{aligned} \alpha&=aa\\ \beta&=bbbbb\\ \gamma&=g
  \end{aligned}
  \qquad\text{versus}\qquad
  \begin{aligned}[t] \delta&=dd\\ \eta&=eeeeee\\ \varphi&=f
  \end{aligned}
  \qquad\text{versus}\qquad
  \begin{gathered}[b] s= x+y\\ d= u - v - w\\ p = x\circ y
  \end{gathered}
\end{equation*}
```

### The cases environment

Although it is possible with standard LaTeX to produce structures of the form

$$P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!\,(-1)^{(r-j)/2} & \text{if } r-j \text{ is even.} \end{cases} \tag{15.8}$$

as demonstrated by a similar example in Section 7.4.1 on page 133, the $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX `cases` environment allows a simpler input:

```
\begin{equation}
 P_{r-j}=\begin{cases} 0 & \text{if $r-j$ is odd,}\\
      r!\,(-1)^{(r-j)/2} & \text{if $r-j$ is even.}
         \end{cases}
\end{equation}
```

There may be more than two cases in the environment, as in the example reproduced here from page 133:

$$y = \begin{cases} -1 & : \quad x<0 \\ \hfill 0 & : \quad x=0 \\ +1 & : \quad x>0 \end{cases}$$

```
\[ y = \begin{cases} -1 &:\quad x<0\\
  \hfill 0 &:\quad x=0\\ +1 &:\quad x>0
       \end{cases} \]
```

## 15.2.7 Equation numbering

### Numbering hierarchy

With the standard LaTeX classes `book` and `report`, equations are given a double number with the chapter designation and then a sequential number starting at 1 for each new chapter. For the `article` class, the equations are numbered sequentially throughout the work.

With the `amsmath` package, it is possible to alter this hierarchy. For example, if an article is to have the equations numbered within each section, with the section number give

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

to redefine the equation numbers to include the section number and to make the `equation` counter reset every time the `section` counter is incremented. This is as though the `equation` counter had been created with `\newcounter{equation}[section]` (Section 10.1.2), something that the user cannot normally bring about. Furthermore, `\theequation` is redefined to be `\thesection.\arabic{equation}`, something that is in the user's power but is not much use if the `equation` counter is never reset.

### Subnumbering equations

On page 190 we give an example of how equations may be subnumbered; that is, the main equation number stays the same and a letter is appended to it, as 1.8a, 1.8b, 1.8c.... The `amsmath` package provides this feature with the `subequations` environment. Numbered equations appearing within

```
\begin{subequations}  ...  \end{subequations}
```

will all have the same main number, which is one more than that of the previous one, with sequential, lowercase letters attached.

Within the environment, the `equation` counter refers to the subnumber, that is, to the letters, while the main number is to be found in the `parentequation` counter. To change the format of the subnumber, say, to 1.8-A, 1.8-B, ..., give

```
\begin{subequations}
\renewcommand{\theequation}
          {\theparentequation-\Alph{equation}}
. . .
\end{subequations}
```

### Referencing equation numbers

The LaTeX cross-reference system is described in Section 11.2.1 and works exactly the same way with AMS-LaTeX: When `\label{`*marker*`}` is issued in a mathematical formula that receives an automatic equation number, that number can be printed anywhere in the text with `\ref{`*marker*`}`, where *marker* is arbitrary text to identify that equation.

The `amsmath` package adds a command `\eqref{`*marker*`}` to print the equation number in parentheses, as it appears beside the math formula. For example, the cases equation on page 288 is referred to as equation 15.8 with `\ref` or as equation (15.8) with `\eqref`.

If `\label` is given immediately after the start of a `subequations` environment, the corresponding `\ref` commands will print the main equation number without the extra letter. In this way one can refer to the entire group of equations. Later `\label` commands are associated with individual equation lines and reference them with the letters.

### Page breaks within multiline formulas

Unlike the standard LaTeX `eqnarray` environment, the AMS-LaTeX multiline math environments do not normally allow any page breaks to occur within them. The idea is that the author should have more control over where such breaks may occur. To allow or force a page break within a multiline equation, one gives

`\displaybreak[`*num*`]`

just before the line-breaking command \\. Here the optional *num* has the same meaning as for the standard `\pagebreak` command (page 36): Without it, a new page is forced, but it may take values of 0–4 to allow a break with increasing degrees of encouragement, whereby 4 also forces the page break.

Alternatively, one can issue `\allowdisplaybreaks` in the document preamble to allow LaTeX to break pages automatically within multiline formulas as necessary. This command also takes an optional argument *num* with possible values between 0 and 4, which make it progressively easier for automatic page breaks to occur.

Once `\allowdisplaybreaks` has been given in the preamble, it is still possible to suppress page breaks within a formula by ending an equation line with * instead of with \\.

### 15.2.8 Package options for amsmath

The main `amsmath` package for $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX recognizes a number of options that may be given when it is loaded with

`\usepackage[`*options*`]{amsmath}`

They are listed here as pairs with opposing effects. The member of each pair that is assumed if neither is given, the default, is indicated by underlining.

<u>`centertags`</u> | `tbtags` The equation number for a `split` environment (page 283) is centered vertically by default. With `tbtags`, it is placed either to the left of the first line or to the right of the last line, depending on the side on which numbers are to appear.

<u>`sumlimits`</u> | `nosumlimits` In displayed formulas, initial and final limits appear below and above the $\sum$ sign with the `sumlimits` option. With `nosumlimits`, they are placed beside the sign, raised and lowered with the usual ^ and _ characters.

Other symbols that are affected by these options are $\sum \prod \coprod \bigcup \biguplus \bigcap \bigsqcup \bigvee \bigwedge \bigodot \bigotimes$ and $\bigoplus$. On the other hand, integral signs are not influenced by them.

`sumlimits`

$$\sum_{n=0}^{\infty} \frac{1}{2^n} = 2; \prod_{i=0}^{m-1} n - i = \frac{n!}{(n-m)!}$$

`nosumlimits`

$$\sum\nolimits_{n=0}^{\infty} \frac{1}{2^n} = 2; \prod\nolimits_{i=0}^{m-1} n - i = \frac{n!}{(n-m)!}$$

The input text is the same for both of the above cases.

`intlimits` | <u>`nointlimits`</u> Integral signs normally have their limits at the side; these options allow them to be placed above and below as for summations.

`intlimits`

$$\int\limits_0^a \sqrt{a^2 - x^2}\,\mathrm{d}x = \int\limits_0^1 a^2\sqrt{1-\sin^2 t}\,\mathrm{d}\sin t = a^2\int\limits_0^{\pi/2} \cos^2 t\,\mathrm{d}t = \frac{\pi a^2}{4}$$

nointlimits

$$\int_0^a \sqrt{a^2 - x^2}\,\mathrm{d}x = \int_0^1 a^2\sqrt{1-\sin^2 t}\,\mathrm{d}\sin t = a^2\int_0^{\pi/2}\cos^2 t\,\mathrm{d}t = \frac{\pi a^2}{4}$$

where again the input text is the same for both cases. As a reminder, the standard LaTeX treatment of limits on integral signs is the same as for `nointlimits`.

`namelimits` | `nonamelimits` The functions `\det`, `\gcd`, `\inf`, `\lim`, `\liminf`, `\limsup` `\max`, `\min`, `\Pr`, and `\sup` frequently take lower limits that normally appear below the name in displayed formulas. With `nonamelimits`, they are placed at the lower right.

namelimits

$$\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x = \mathrm{e} = 2.7182\ldots$$

nonamelimits

$$\lim\nolimits_{x\to\infty}\left(1+\frac{1}{x}\right)^x = \mathrm{e} = 2.7182\ldots$$

The above options determine the standard placement of limits for the entire document. It is still possible to change the behavior in any particular case with the `\limits` and `\nolimits` commands, as in normal LaTeX.

The remaining package options select the side for equation numbers and the horizontal positioning of equations.

`leqno` | `reqno` The standard location for equation numbers is on the right side, at the margin; with `leqno`, they are placed on the left side of the equation. For the $\mathcal{AMS}$ classes, the default is `leqno`.

`fleqn` With this option, all displayed equations are printed flush left, set off from the left margin by an amount `\mathindent`. Without this option, equations are centered. (This is like the `fleqn` class option for the standard classes, page 43.)

## 15.3 Further $\mathcal{AMS}$-LaTeX packages

The $\mathcal{AMS}$-LaTeX packages described in this section must be loaded explicitly if their features are to be exploited. Unlike the `amsbsy` and `amstext` packages, they are not loaded automatically with `amsmath`. They may, however, be used on their own, independently of the main package.

### 15.3.1 Extended theorem declarations

**Package:** `amsthm`

The `amsthm` package offers many additional possibilities for generating theorem-like declarations described for standard LaTeX in Section 4.5.

As for standard LaTeX, a new theorem declaration is created with a statement like

```
\newtheorem{com}{Comment}
```

where the first argument is the name of the theorem type (here `com`) and the second is the title that is printed when the theorem declaration is invoked. For example,

```
\begin{com}
   Theorem declarations can have any name.
\end{com}
```

produces the declaratory text

**Comment 1.** *Theorem declarations can have any name.*

In addition to the two mandatory arguments, the `\newtheorem` command may have one of two optional ones. The complete syntax is

`\newtheorem{`*type*`}[`*num_like*`]{`*title*`}`
or
`\newtheorem{`*type*`}{`*title*`}[`*in_counter*`]`

where *num_like* is the name of an existing theorem-like declaration, which is to be numbered in the same sequence as *type*, and *in_counter* is a counter name like `chapter` or `section` to reset the numbers of the *type* declarations.

All this is standard LaTeX so far. The `amsthm` package adds the following features:

- A `\newtheorem*` is provided that defines an unnumbered theorem structure.
- Three predefined theorem styles are available:
  - `plain`, in which the title and number are in boldface and the text italic;
  - `definition`, with title and number in boldface and the text in normal font;
  - `remark`, for title and number in italic and the text normal.

  The desired style is activated by first issuing `\theoremstyle{`*style*`}`; all subsequent `\newtheorem` statements will have this style until a new one is activated.
- A `\swapnumbers` can be issued to cause all following new theorem types to have the numbers appear before the title, as **1 Comment.**
- New theorem styles may be defined by means of the `\newtheoremstyle` command, or additional predefined styles may be loaded with package options. Since this is fairly specialized and complex, it is best to examine the example file `thmtest.tex` or read the documentation in `amsthm.dtx`.
- A `proof` environment is available for presenting short proofs. It is an unnumbered structure with the title *Proof*. The text is terminated automatically with the Q.E.D. symbol □. This symbol may be altered by redefining the command `\qedsymbol`; it may be printed at any time by issuing `\qed`.

  There is also a `\qedhere` command to be given within a displayed equation or a list at the end of the proof. This moves the symbol to the end of that line; otherwise it appears below the displayed equation or list, and possibly on the next page.

The amsthm package has much in common with Frank Mittelbach's theorem package in the tools collection of Section B.5.4.

### 15.3.2 Commutative diagrams

**Package:** amscd

The extra $\mathcal{AMS}$-LaTeX package amscd makes it easier to generate commutative diagrams like the one here at the right.

$$\begin{CD} S^{\mathcal{W}_\Lambda}\otimes T @>j>> T\\ @VVV @VV{\operatorname{End} P}V\\ (S\otimes T)/I @= (Z\otimes T)/J \end{CD}$$

These diagrams are created within the CD environment using some additional arrow commands. These bear the rather unusual names: @>>> @<<< @AAA and @VVV for arrows pointing right, left, upwards, and downwards, respectively. The command @= draws a horizontal double rule, a lengthened equals sign.

Any text or symbols between the first and second > or < characters will appear above the horizontal arrow in \scriptstyle font. Similarly, any text or symbols between the second and third characters will be printed below the arrow.

For vertical arrows, text or symbols between the first and second A or V are placed to the left; those between the second and third to the right, again in \scriptstyle.

The above example diagram, taken from the $\mathcal{AMS}$-LaTeX manual amsldoc.tex, was produced with

```
\[ \begin{CD}
   S^{\mathcal{W}_\Lambda}\otimes T @>j>>  T\\
   @VVV                             @VV{\End P}V\\
   (S\otimes T)/I                @= (Z\otimes T)/J
   \end{CD}  \]
```

The command \End to print the function name 'End' is not standard. It must be previously defined with \DeclareMathOperator{\End}{End} (see page 279).

### 15.3.3 References with upref package

**Package:** upref

Normally the numbers printed with the \ref and \pageref commands are in the current font, whether that be bold, italic, or upright. To ensure that the numbers are always upright, load the extra $\mathcal{AMS}$-LaTeX package upref.

## 15.4 The $\mathcal{AMS}$ fonts

The $\mathcal{AMS}$ makes a number of fonts available to complement the regular Computer Modern fonts provided with the standard TeX/LaTeX installation. They include extra math alphabets, supplemental CM bold math italic and symbol fonts in smaller sizes than 10 pt, Cyrillic fonts, and additional symbol fonts.

In the next sections we describe these various fonts and how to take advantage of them.

### 15.4.1 Extra CM math fonts

Standard TeX installations of Computer Modern fonts provide bold math italic `cmmib10`, the bold symbols `cmbsy10`, and math extensions `cmex10` fonts only in 10 pt size, as indicated by the suffix 10 to their names. The AMS has supplemented these with versions in sizes 5–9 pt.

| | | | | |
|---|---|---|---|---|
| cmmib5 | cmmib6 | cmmib7 | cmmib8 | cmmib9 |
| cmbsy5 | cmbsy6 | cmbsy7 | cmbsy8 | cmbsy9 |
| | | cmex7 | cmex8 | cmex9 |

The small caps font `cmcsc10` is also given companions `cmcsc8` and `cmcsc9`.

The normal LaTeX installation automatically assumes that these fonts are on the system and incorporates them into the necessary NFSS font definition files. Substitutions will be made if they are missing.

### 15.4.2 Cyrillic fonts

The AMS Cyrillic fonts were originally used in reviews of books published in Russian and other Slavic languages in which the titles were to be rendered in the original language. In 1988, the Humanities and Arts Computing Center of the University of Washington redesigned them for general-purpose Slavic studies, adding the pre-Revolutionary and accented letters. The overall appearance was also greatly enhanced.

These fonts all bear the prefix `wncy`, followed by the style designation `r` (upright), `b` (bold), `i` (italic), `sc` (small caps), or `ss` (upright sans serif), and the design size in points.

The best way to enable the Cyrillic fonts is simply to select font encoding `OT2` under NFSS, as illustrated in Section A.2 on page 365. The `.fd` (font definition) files for the Cyrillic fonts have been set up such that the other font attributes fully parallel those of the Latin fonts. This means `wncyr10` has all the same attributes as `cmr10`, except for the encoding: family `cmr`, shape `n`, series `m`. (Of course, if the CM fonts are not the current standard ones, it will require more than just selecting the `OT2` encoding to activate these Cyrillic fonts.)

The layout of the Cyrillic fonts has been chosen in such a way that the input text may be entered following the regular English transliteration scheme. Thus Cyrillic С is in position 83 where Latin S is normally situated. Typing S when a Cyrillic font is active outputs the correct equivalent С. Numerals and punctuation are to be found in the standard locations, so they may be typed in as usual. 'Санкт-Петербург 10?' is thus generated by `{\cyr Sankt-Peterburg 10?}`.

Since the Cyrillic alphabet possesses more letters than the Latin, many of them must be transliterated with multiletter combinations. These are automatically programmed into the fonts using TeX's ligature system. For example, `Ch` is treated as a ligature for symbol 81 Ч just as `fi` is for *fi* in a Latin font. This means that multiletter transliterations are simply typed in. The input for 'Хрущев' is `{\cyr Khrushchev}`, where `Kh` → Х and `shch` → щ. The transliteration scheme is that for English; other languages have their own systems to reproduce the original pronunciation. For example, 'Горбачёв' is *Gorbatschow* in German, *Gorbaciov* in Italian, and *Gorbachev* in English. These other schemes do *not* work with these fonts.

Not all letters can be produced so automatically (for example, the ё in Горбачёв), and for this reason the $\mathcal{AMS}$ provides a file `cyracc.def` containing macro definitions for accented letters and other special features, such as the hard and soft signs. When these macros are given in a Latin font, additional transliteration symbols appear.

### 15.4.3 Extra math symbols

**Package:** `amsfonts` `amssymb`

The set of symbols in the CM math symbol fonts by no means exhausts the fantasies of active mathematicians. To overcome this deficiency, the $\mathcal{AMS}$ has produced two fonts, `msam10` and `msbm10`, containing only symbols, arrows, and the blackboard characters. They are also available in sizes 5–9 pt. Since these fonts originated in the days when TeX could only handle 128 characters in any font, that is exactly how many they contain. Today they could be combined into one font of 256 characters.

Two packages permit access to this treasure trove of hieroglyphs:

`amsfonts` enables the `\mathbb` math alphabet command for the blackboard characters, and defines those symbol names that are otherwise only provided in the `latexsym` package (Section 7.3.3).

`amssymb` is the more convenient package, which loads `amsfonts` and then defines names for all the symbols in the two fonts.

For example,

```
\[ \circlearrowright \Cup \lessapprox \lll \varpropto \because
   \circeq \vDash \blacktriangle \sphericalangle \]
```

$$\circlearrowright \quad \Cup \quad \lessapprox \quad \lll \quad \varpropto \quad \because \quad \circeq \quad \vDash \quad \blacktriangle \quad \sphericalangle$$

The 'blackboard' characters are selected with the math alphabet command `\mathbb`. Thus, `$\mathbb{A B C ..}$` produces

$\mathbb{A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ N\ M\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z}$

All the symbols and their associated names from the `amssymb` package are to be found in Tables G.20 through G.26 on pages 551–553.

### 15.4.4 Euler fonts

The *Euler* fonts, named after the eighteenth-century mathematician Leonhard Euler, were designed by Hermann Zapf. Their main purpose in mathematics is to be a substitute for the CM calligraphic math alphabets.

**Package:** `eucal`

The package `eucal` redefines the `\mathcal` command to use the Euler script characters in place of the CM calligraphic letters. If this package is loaded with the option `mathscr`, the command `\mathcal` is left unchanged and, instead, `\mathscr` is defined to invoke these letters. In this case, `$\mathscr{A B C ..}$` produces

$$\mathscr{A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ N\ M\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z}$$

**Package:** `eufrak`

On the other hand, the package `eufrak` defines the math alphabet command `\mathfrak`, with which `$\mathfrak{A B C ..}$` yields

$$\mathfrak{A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ N\ M\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z}$$

This math alphabet is also enabled with the `amsfonts` package.