

Resource Saving Minimalism



Tools
Scientifically Approved

pjs64



Blog-roller ·
pjs64.wordpress.com

No-Kite-Surfer ·

Org-mode Rookie
2022 Somewhere

Contents

I	Form Graphs	5
1.	Preface for the HTML Version	5
2.	Introduction	5
3.	Options for the <code>amsmath</code> package	7
4.	Displayed equations	8
4.1.	Introduction	8
4.2.	Single equations	10
4.3.	Split equations without alignment	10
4.4.	Split equations with alignment	12
4.5.	Equation groups without alignment	12
4.6.	Equation groups with mutual alignment	13
4.7.	Alignment building blocks	15
4.8.	Adjusting tag placement	16
4.9.	Vertical spacing and page breaks in multiline displays . .	17
4.10.	Interrupting a display	18
4.11.	Equation numbering	19
4.11.1.	Numbering hierarchy	19
4.11.2.	Cross references to equation numbers	19
4.11.3.	Subordinate numbering sequences	20
4.11.4.	Numbering style	21
5.	Miscellaneous mathematical features	21
5.1.	Matrices	21
5.2.	Math spacing commands	23
5.3.	Dots	24
5.4.	Nonbreaking dashes	24
5.5.	Accents in math	25
5.6.	Roots	25
5.7.	Boxed formulas	26

5.8. Over and under arrows	26
5.9. Extensible arrows	26
5.10. Affixing symbols to other symbols	27
6. Text in Math environment	27

Part I

Form Graphs

1 Preface for the HTML Version

This script is an exercise of `amsmath` features, and a test for their compatibility with ORG MODE export to HTML. I try to exactly reproduce the User Manual.

Issues related to the collaboration of ORG MODE, LATEX, and MATHJAX are addressed in the text; see the mathjax [discussion about ams](#) in the [extension list](#) for LATEX. I'm plan to separate these parts from the mathematical content.

Medium.com and the free version of WORDPRESS don't accept JAVASCRIPT inclusion, so MATHJAX is not an option. For kind of an ugly exception handling I can use the PNG export of LATEX math by setting the option `tex:dvipng`. The side effects – and out-ruling arguments – of this surrogate “depiction” begin to show in the examples of Section 4.6 *Equation groups with mutual alignment*:

- internal label references aren't resolvable
- equation arrays which are supposed to have different numbers get only one number.
- The equation numbers don't have parentheses.

So, the `tex:dvipng` solution has to be revisited for being a valuable solution. Another reason to change the system.

2 Introduction

The `amsmath` package is a LATEX package that provides miscellaneous enhancements for improving the information structure and printed output of documents that contain mathematical formulas. Readers unfamiliar with LATEX should consult [1]. If you have an up-to-date version of LATEX, the `amsmath` package is normally provided along with it. Upgrading when a newer version of the `amsmath` package is released can be done via the [download link](#).

This documentation describes the features of the `amsmath` package and discusses how they are intended to be used. It also covers some ancillary packages, like `amsbsy`, `amsopn`, `amsxtra`, `amscd`, or `amstext`.

These all have something to do with the contents of math formulas. For information on extra math symbols and math fonts, see *AMS fonts* at `ams.org`. For documentation of the `amsthm` package or AMS document classes, like `amsart` or `amsbook`, see `amsthm` or Section *AMS Author Handbook* in the *Author Resource Center* at `ams.org`.

If you are a long-time LATEX user and have lots of mathematics in what you write, then you may recognize solutions for some familiar problems in this list of `amsmath` features:

- A convenient way to define new ‘operator name’ commands analogous to `\sin` and `\lim`, including proper side spacing and automatic selection of the correct font style and size (even when used in sub- or superscripts).
- Multiple substitutes for the `eqnarray` environment to make various kinds of equation arrangements easier to write.
- Equation numbers automatically adjust up or down to avoid overprinting on the equation contents (unlike `eqnarray`).
- Spacing around equals signs matches the normal spacing in the `equation` environment (unlike `eqnarray`).
- A way to produce multiline subscripts as are often used with summation or product symbols.
- An easy way to substitute a variant equation number for a given equation instead of the automatically supplied number.
- An easy way to produce subordinate equation numbers of the form (1.3a) (1.3b) (1.3c) for selected groups of equations.

The `amsmath` package is distributed together with some small auxiliary packages:

- `amsmath` – Primary package, provides various features for displayed equations and other mathematical constructs.
- `amstext` – Provides a `\text{}` command for typesetting a fragment of text inside a display.
- `amsopn` – Provides `\DeclareMathOperator{}` for defining new ‘operator names’ like `\sin` and `\lim`.
- `amsbsy` – For backward compatibility this package continues to exist but use of the newer `bm` package that comes with LATEX is

recommended instead.

- `amscd` – Provides a CD environment for simple commutative diagrams (no support for diagonal arrows).
- `amsxtra` – [obsolete, use `amscd`] Provides certain odds and ends such as `\fracwithdelims{}` and `\accentedsymbol{}`, for compatibility with documents created using version 1.1.

The `amsmath` package incorporates `amstext`, `amsopn`, and `amsbsy`. The features of `amscd` and `amsxtra`, however, are available only by invoking those packages separately.

The independent `mathtools` package provides some enhancements to `amsmath` and loads `amsmath` automatically. Some `mathtools` facilities will be noted below as appropriate.

3 Options for the `amsmath` package

The `amsmath` package has the following options:

- `centertags` – (default) For an equation containing a `split` environment, place equation numbers vertically centered on the total height of the equation.
- `tbtags` – ‘Top-or-bottom tags’: For an equation containing a `split` environment, place equation numbers level with the last (resp. first) line, if numbers are on the right (resp. left).
- `sumlimits` – (default) Place the subscripts and superscripts of summation symbols above and below, in displayed equations. This option also affects other symbols of the same type — \prod , \coprod , \otimes , \oplus , and so forth — but excluding integrals (see below).
- `nosumlimits` – Always place the subscripts and superscripts of summation-type symbols to the side, even in displayed equations.
- `intllimits` – Like `sumlimits`, but for integral symbols.
- `nointlimits` – (default) Opposite of `intllimits`.
- `namelimits` – (default) Like `sumlimits`, but for certain ‘operator names’ such as `det`, `inf`, `lim`, `max`, `min`, that traditionally have subscripts placed underneath when they occur in a displayed equation.
- `nonamelimits` – Opposite of `namelimits`.
- `alignedleftspaceyes` –

- `alignedleftspaceno` –
- `alignedleftspaceyesifneg` –

To use one of these package options, put the option name in the optional argument of the `\usepackage{}` command — e.g.,

For AMS document classes and any other classes that preload `amsmath` desired options must be specified with the `\documentclass{}`— e.g.,

The `amsmath` package also recognizes the following options which are normally selected (implicitly or explicitly) through the `\documentclass{}` command, and thus need not be repeated in the option list of the `\usepackage{amsmath}` statement.

- `leqno` – Place equation numbers on the left.
- `reqno` – Place equation numbers on the right.
- `fleqn` – Position equations at a fixed indent from the left margin rather than centered in the text column.

Three options have been added to control the space to the left of `aligned` and `gathered` environments. Prior to the 2017 release a thin space was added to the left but not the right of these constructs. This appears to have been an accidental feature of the implementation and was typically corrected by prefixing the environments with `\!`.

The new default behavior is aimed to ensure that the environments do not have a thin space added in most cases, and that existing documents using `\!\begin{aligned}` continue to work as before.

- `alignedleftspaceyes` – Always add `\,` to the left of `aligned` and `gathered`.
- `alignedleftspaceno` – Never add `\,` to the left of `aligned` and `gathered`.
- `alignedleftspaceyesifneg` – Only add `\,` if the environment is prefixed by negative space. (New default behavior.)

4 Displayed equations

4.1 Introduction

The `amsmath` package provides a number of additional displayed equation structures beyond the ones provided in basic LATEX. The

augmented set includes `align`, `alignat`, `equation`, `flalign`, `gather`, `multline`, with asterisk alternative, and `split` without.



Although the standard `eqnarray` environment remains available, it is better to use `align` or `equation` & `split` instead. Within `eqnarray`, spacing around signs of relation is not the preferred mathematical spacing, and is inconsistent with that spacing as it appears in other environments. Long lines in this environment may result in misplaced or overprinted equation numbers. This environment also does not support the use of `\qed` or `\qedhere` as provided by theorem packages.

Except for `split`, each environment has both starred and unstarred forms, where the unstarred forms have automatic numbering using L^AT_EX's `equation` counter. You can suppress the number on any particular line by putting `\notag` before the end of that line.



`\notag` should not be used outside a display environment as it will mess up the numbering.

You can also override a number with a tag of your own using `\tag{label}`, where `label` means arbitrary text such as `$$` or `ii` used to “number” the equation. A tag can reference a different tagged display by use of `\tag{\ref{label}modifier}` where `modifier` is optional. If you include `hyperref`, use `\ref*{}`; using the starred form of `\ref{}` prevents a reference to a modified tag containing a nested link from linking to the original display.

There is also a `\tag*{}` command that causes the text you supply to be typeset literally, without adding parentheses around it. `\tag{}` and `\tag*{}` can also be used within the unnumbered versions of all the `amsmath` alignment structures. Some examples of the use of `\tag{}` may be found in the sample files `testmath.tex` and `subeqn.tex` provided with the `amsmath` package.

The `split` environment is a special subordinate form that is used only *inside* one of the others. It cannot be used inside `multline`, how-

ever. `split` supports only one alignment column, i.e. only one `&` character; if more are needed, `aligned` or `alignedat` should be used. The width of a `split` structure is the full line width.

In the structures that do alignment, i.e., `split`, `align`, `alignat`, and `flalign`, relation symbols have an `&` before them but not after — unlike `eqnarray`. Putting the `&` after the relation symbol will interfere with the normal spacing; it has to go before.

In all multiline environments, lines are divided by `\\`.



The `\\` should *not* be used to end the last line. Using it there will result in unwanted extra vertical space following the display.



In *all* math environments, no matter if inline or display, blank lines, or the equivalents to `\par`, are not permitted, and will result in an error. → See Section 4.9.

4.2 Single equations

The `equation` environment is for a single equation with an automatically generated number. The `equation*` environment is the same except for omitting the number.



Basic LATEX doesn't provide an `equation*` environment, but rather a functionally equivalent environment named `displaymath`.

The wrapper `\[... \]` is equivalent to `equation*`.

4.3 Split equations without alignment

The `multline` environment is a variation of the `equation` environment used for equations that don't fit on a single line. The first

<pre>\begin{equation*} a=b \end{equation*}</pre>	$a = b$	$(1) \quad a = b$	<pre>\begin{equation} a=b \end{equation}</pre>
--	---------	-------------------	--

Fig. 1 – One line math

line of a `multline` will be at the left margin and the last line at the right margin, except for an indentation on both sides in the amount of `\multlinegap{}`. Any additional lines in between will be centered independently within the display width, unless the `fleqn` option is in effect.

<pre>\begin{equation}\label{xx} \begin{split} a& =b+c-d\\ & \quad +e-f\\ & =g+h\\ & =i \end{split} \end{equation}</pre>	<pre>\begin{multline} a =b+c-d\\ +e-f\\ =g+h\\ =i \end{multline}</pre>
$(2) \quad \left \begin{array}{l} a = b + c - d \\ + e - f \\ = g + h \\ = i \end{array} \right.$	$\left \begin{array}{l} a = b + c - d \\ + e - f \\ = g + h \\ = i(3) \end{array} \right.$

Fig. 2 – Equation & split versus multiline.

Like `equation`, `multline` has only a single equation number (thus, none of the individual lines should be marked with `\notag`). The equation number is placed on the last line (`reqno` option) or first line (`leqno` option); vertical centering as for `split` is not supported by `multline`.%

It's possible to force one of the middle lines to the left or right with commands `\shoveleft{}`, `\shoveright{}`. These commands take the entire line as an argument, up to but not including the final `\`; for example `\shoveright{+e-f}\` would shift the line `\(+e-f` in Figure 2 to the right leaving a virtual gap for the number (3) below.

The value of `\multlinegap{}` can be changed with the LATEX core commands `\setlength{}` or `\addtolength{}`.

4.4 Split equations with alignment

Like `multline`, the `split` environment is for *single* equations that are too long to fit on one line and hence must be split into multiple lines. Unlike `multline`, however, the `split` environment provides for alignment among the split lines, using `&` to mark alignment points. Unlike the other `amsmath` equation structures, the `split` environment provides no numbering, because it is intended to be used *only inside some other displayed equation structure*, usually an `equation`, `align`, or `gather` environment, which provides the numbering. Check the difference in Figure 2.

The `split` structure should constitute the entire body of the enclosing structure, apart from commands like `\label{}` that produce no visible material.

4.5 Equation groups without alignment

The `gather` environment is used for a group of consecutive equations when there is no alignment desired among them; each one is centered separately within the text width, see Figure 3.

```
\begin{gather}
a_1=b_1+c_1\\
a_2=b_2+c_2-d_2+e_2
\end{gather}
```

$$\left. \begin{array}{l} (4) \quad a_1 = b_1 + c_1 \\ (5) \quad a_2 = b_2 + c_2 - d_2 + e_2 \end{array} \right|$$

Fig. 3 – A minimal example of two equations gathered together in the `gather` `amsmath` environment

Equations inside `gather` are separated by a `\\` command. Any equation in a `gather` may mix with `split` environments, for example:

```
\begin{gather} 1st equation \\ 2nd equation \\ \begin{split} 3rd & two \\ & & line eqn \end{split} \\ 4th equation \\ 5th ... get the picture ... \end{gather}
```

4.6 Equation groups with mutual alignment

The `align` environment is used for two or more equations when vertical alignment is desired; usually binary relations such as equal signs are aligned, like in Figure 4.

$\begin{aligned} (6) \quad a_1 &= b_1 + c_1 \\ (7) \quad a_2 &= b_2 + c_2 - d_2 + e_2 \end{aligned}$	$\begin{aligned} (8) \quad a_{11} &= b_{11} & a_{12} &= b_{12} \\ (9) \quad a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} \end{aligned}$
--	---


```
\begin{align}
a_{11}&=b_{11}&
a_{12}&=b_{12} \\
a_{21}&=b_{21}&
a_{22}&=b_{22}+c_{22}
\end{align}
```

```
\begin{align}
a_1&=b_1+c_1 \\
a_2&=b_2+c_2-d_2+e_2
\end{align}
```

Fig. 4 – Alignment of many equations

To have several equation columns side-by-side, use extra ampersands to separate the columns:

$x = y$	$X = Y$	$a = b + c$	(1)
$x' = y'$	$X' = Y'$	$a' = b$	(2)
$x + x' = y + y'$	$X + X' = Y + Y'$	$a'b = c'b$	(3)

```
\begin{align}
x&=y & & X&=Y & & a&=b+c \\
x'&=y' & & X'&=Y' & & a'&=b \\
x+x'&=y+y' & & X+X'&=Y+Y' & & a'b&=c'b
\end{align}
```

Line-by-line annotations on an equation can be done by judicious application of `\text{}` inside an `align` environment:

$x = y_1 - y_2 + y_3 - y_5 + \dots$	by (1)	(4)
$= y' \circ y^*$	by (2)	(5)
$= y(0)y'$	by Axiom 1.	(6)

```

\begin{align}
x& = y_1 - y_2 + y_3 - y_5 + \dots \quad \&\& \text{\textit{by \eqref{eq:C}}}\ \\
& = y' \circledast y^* \quad \&\& \text{\textit{by \eqref{eq:D}}}\ \\
& = y(0) y' \quad \&\& \text{\textit{by Axiom 1.}}
\end{align}

```

A variant environment `alignat` allows the horizontal space between equations to be explicitly specified. This environment takes one argument, the number of “equation columns” (the number of pairs of right-left aligned columns; the argument is the number of pairs): count the maximum number of ampersands in any row, add 1 and divide by 2.

$$x = y_1 - y_2 + y_3 - y_5 + \dots \quad \text{by (1)} \quad (7)$$

$$= y' \circledast y^* \quad \text{by (2)} \quad (8)$$

$$= y(0)y' \quad \text{by Axiom 1.} \quad (9)$$

```

\begin{alignat}{2}
x& = y_1 - y_2 + y_3 - y_5 + \dots \quad \&\& \text{\textit{by \eqref{eq:C}}}\ \\
& = y' \circledast y^* \quad \&\& \text{\textit{by \eqref{eq:D}}}\ \\
& = y(0) y' \quad \&\& \text{\textit{by Axiom 1.}}
\end{alignat}

```

The environment `flalign`, i.e., full length alignment, stretches the space between the equation columns to the maximum possible width, leaving only enough space at the margin for the equation number, if present.

(3.21) $x = y$	<code>\begin{flalign}</code>	$X = Y$
(3.22) $x' = y'$	<code>x&=y & X&=Y\ \\</code>	$X' = Y'$
(3.23) $x + x' = y + y'$	<code>x'&=y' & X'&=Y' \\</code>	$X + X' = Y + Y'$
	<code>x+x'&=y+y' & X+X'&=Y+Y' \\</code>	
	<code>\end{flalign}</code>	
$x = y$	<code>\begin{flalign*}</code>	$X = Y$
$x' = y'$	same as above	$X' = Y'$
$x + x' = y + y'$	<code>\end{flalign*}</code>	$X + X' = Y + Y'$

Fig. 5 – Alignment of many equations

4.7 Alignment building blocks

Like `equation`, the multi-equation environments `gather`, `align`, and `alignat` are designed to produce a structure whose width is the full line width. This means, for example, that one cannot readily add parentheses around the entire structure. But the variants `gathered`, `aligned`, and `alignedat` are provided whose total width is the actual width of the contents; thus they can be used as a component in a containing expression. E.g.,

$$\left. \begin{aligned} \frac{\partial B}{\partial t} &= -\nabla \times E, \\ \frac{\partial E}{\partial t} &= \nabla \times B - 4\pi j, \end{aligned} \right\} \text{Maxwell's equations}$$

```
\begin{equation*}\left.\begin{aligned}
\frac{\partial B}{\partial t} &= -\nabla \times E, \\
\frac{\partial E}{\partial t} &= \nabla \times B - 4\pi j,
\end{aligned}\right\} \text{Maxwell's equations} \end{equation*}
```

Like the `array` environment, these `-ed` variants also take an optional `[t]` top, `[b]` bottom or the default `[c]` center option argument to specify vertical positioning.



For maximum interoperability, do not insert a space or line break before the option.



Certain equation environments wrap their contents in an unbreakable box, with the consequence that neither `\displaybreak` nor `\allowdisplaybreaks` will have any effect on them. These include `split`, `aligned`, `gathered`, and `alignedat`.

Case constructions like the following are common in mathematics:

$$P_{r-j} = \begin{cases} 0 & \text{if } r - j \text{ is odd,} \\ r! (-1)^{(r-j)/2} & \text{if } r - j \text{ is even.} \end{cases} \quad (10)$$

and in the `amsmath` package there is a `cases` environment to make them easy to write:

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

Notice the use of `\text{}` – → see Section 6 – and the nested math formulas. The `cases` environment is set in `\textstyle`. If `\displaystyle` is wanted, it must be requested explicitly; `mathtools` provides a `dcases` environment for this purpose.

The `-ed` and `cases` environments must appear within an enclosing math environment, which can be either in text, between \dots , or in any of the display environments.

4.8 Adjusting tag placement

Placing equation numbers can be a rather complex problem in multiline displays. The environments of the `amsmath` package try hard to avoid overprinting an equation number on the equation contents, if necessary moving the number down or up to a separate line. Difficulties in accurately calculating the profile of an equation can occasionally result in number movement that doesn't look right.

A `\raisetag{}` command is provided to adjust the vertical position of the current equation number, if it has been shifted away from its normal position. To move a particular number up by six points, write `\raisetag{6pt}`.



At the end of a display, the `\raisetag{}` command also shifts up the text following the display.



A `\raisetag{}` adjustment is fine tuning like line breaks and page breaks, and should be left undone until your document is nearly finalized, or you may end up redoing the fine tuning several times to keep up with changing document contents.

4.9 Vertical spacing and page breaks in multiline displays

You can use the `\[dimension]` command to get extra vertical space between lines in all the `amsmath` displayed equation environments, as is usual in LATEX.



Do not type a space between the `\` and the following `[`; *only* for display environments defined by `amsmath` the space is interpreted to mean that the bracketed material is part of the document content.

When the `amsmath` package is in use between equation lines are normally disallowed; the philosophy is that page breaks in such material should receive individual attention from the author. To get an individual page break inside a particular displayed equation, a `\displaybreak` command is provided. `\displaybreak` is best placed immediately before the `\` where it is to take effect. Like LATEX's `\pagebreak`, `\displaybreak` takes an optional argument between 0 and 4 denoting the desirability of the pagebreak. `\displaybreak[0]` means “it is permissible to break here” without encouraging a break; `\displaybreak` with no optional argument is the same as `\displaybreak[4]` and forces a break.

If you prefer a strategy of letting page breaks fall where they may, even in the middle of a multiline equation, then you might put `\allowdisplaybreaks[1]` in the preamble of your document. An optional argument 1–4 can be used for finer control: [1] means allow page breaks, but avoid them as much as possible; values of 2,3,4 mean increasing permissiveness. When display breaks are enabled with `\allowdisplaybreaks`, the `*` command can be used to prohibit a pagebreak after a given line, as usual.



Certain equation environments wrap their contents in an unbreakable box, with the consequence that neither `\displaybreak` nor `\allowdisplaybreaks` will have any

effect on them. These include `split`, `aligned`, `gathered`, and `alignedat`.

4.10 Interrupting a display

The command `\intertext{}` is used for a short interjection of one or two lines of text in the middle of a multiple-line display structure; → see also Section 6 *The text command* of the *User’s Guide for the `amsmath` Package*. Its salient feature is preservation of the alignment, which would not happen if you simply ended the display and then started it up again afterwards. `\intertext{}` may only appear right after a `\\` or `*` command. Notice the position of the word “and” in the example depicted in Figure 6, produced by

```
\begin{align}
  A_1&=N_0(\lambda;\Omega')-\phi(\lambda;\Omega'),\\
  A_2&=\phi(\lambda;\Omega')-\phi(\lambda;\Omega),\\
\intertext{and}
  A_3&=\mathcal{N}(\lambda;\omega).
\end{align}
```

$$(3.25) \quad A_1 = N_0(\lambda; \Omega') - \phi(\lambda; \Omega'),$$

$$(3.26) \quad A_2 = \phi(\lambda; \Omega') - \phi(\lambda; \Omega),$$

and

$$(3.27) \quad A_3 = \mathcal{N}(\lambda; \omega).$$

```
#+header: :noeval :exports code
#+begin_src latex
\begin{align}
  A_1&=N_0(\lambda;\Omega')-\phi(\lambda;\Omega'),\\
  A_2&=\phi(\lambda;\Omega')-\phi(\lambda;\Omega),\\
\intertext{and}
  A_3&=\mathcal{N}(\lambda;\omega).
\end{align}
#+end_src
```

Fig. 6 – Left: result of the `\intertext{}` interjection of a text paragraph without breaking the equation alignment. – Right: the ORG MODE source code block for inserting the code of the example.



The `mathtools` package provides a command `\shortintertext{}` that is intended for use when the interjected text is only a few words; it uses less vertical space than `\intertext{}`. This is most effective when equation numbers are on the right.

4.11 Equation numbering

4.11.1 Numbering hierarchy

In LATEX if you wanted to have equations numbered within sections — that is, have equation numbers (1.1), (1.2), ..., (2.1), (2.2), ..., in sections 1, 2, and so forth — you could redefine `\theequation{}` as suggested in [1]¹:

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

This works pretty well, except that the equation counter won't be reset to zero at the beginning of a new section or chapter, unless you do it yourself using `\setcounter{}`. To make this a little more convenient, the `amsmath` package provides the command `\numberwithin{ }{ }`. To have equation numbering tied to section numbering, with automatic reset of the equation counter, write

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

As its name implies, the `\numberwithin{ }` command can be applied to any counter, not just the equation counter. Addition of the blog author:

“... but the results may not be satisfactory in all cases because of potential complications. See the discussion of the `\addtoreset` command in Subsection *A.1.4 Defining and changing counters*, Section *A.1 Linking markup and formatting* of Appendix A *LATEX Overview for Preamble, Package, and Class Writers*”

— Subsection *8.2.14 Resetting the equation counter*, Section *8.2 Display and alignment structures for equations* of [2]’s Chapter 8 *Higher Mathematics*

4.11.2 Cross references to equation numbers

To make cross-references to equations easier, an `\eqref{ }` command is provided. This automatically supplies the parentheses around the

¹Sections *6.3 Numbering*, Chapter 6 *Designing It Yourself* and Subsection *C.8.4 Numbering* in Section *C.8 Definitions, Numbering, and Programming*, Appendix C *Reference Manual* of The LATEX Manual [1]

equation number. I.e., if `\ref{abc}` produces 3.2 then `\eqref{abc}` produces (3.2). The parentheses around an `\eqref{}` equation number will be set in upright type regardless of type style of the context.

4.11.3 Subordinate numbering sequences

The `amsmath` package provides also a wrapper environment, `subequations`, to make it easy to number equations in a particular `\align{}` or similar group with a subordinate numbering scheme. For example

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

causes all numbered equations within that part of the document to be numbered (4.9a) (4.9b) (4.9c) ..., if the preceding numbered equation was (4.8). A `\label{}` command immediately after `\begin{subequations}` will produce a `\ref{}` of the parent number 4.9, not 4.9a. The counters used by the `subequations` environment are `parentequation` and `equation` and `\addtocounter{}`, `\setcounter{}`, `\value{}`, etc., can be applied as usual to those counter names. To get anything other than lowercase letters for the subordinate numbers, use standard LATEX methods for changing numbering style; → see, again, Sections *6.3 Numbering*, Chapter *6 Designing It Yourself* and Subsection *C.8.4 Numbering* in Section *C.8 Definitions, Numbering, and Programming*, Appendix *C Reference Manual* of the LATEX manual [1]. For example, redefining `\theequation{}` as follows will produce roman numerals.

```
\begin{subequations}
\renewcommand{\theequation}{\theparentequation \roman{equation}}
```



Note that the `\renewcommand` has to be placed into the `subequations` environment, right behind the `\begin{}` statement.

4.11.4 Numbering style

The default equation number is set in `\normalfont`. This means that in bold section headings, bold is suppressed; a workaround for this is to use the standard `\ref{}` with parentheses rather than `\eqref{}`.

If another font size is specified for a numbered display, the size of the equation number will inherit this size. The default size can be forced throughout a document by applying this patch in the preamble:

```
\makeatletter
\renewcommand{\maketag@@@}[1]{\hbox{\m@th\normalsize\normalfont#1}}%
\makeatother
```

This modification may be included in a future version of `amsmath`.

5 Miscellaneous mathematical features

5.1 Matrices

The `amsmath` package provides some environments for matrices beyond the basic `array` environment of LATEX. The `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` and `Vmatrix` have, respectively, `()`, `[]`, `{}`, `||`, and `|||` delimiters built in. For naming consistency there is a `matrix` environment sans delimiters. This is not entirely redundant with the `array` environment; the matrix environments all use more economical horizontal spacing than the rather prodigal spacing of the `array` environment. Also, unlike the `array` environment, you don't have to give column specifications for any of the matrix environments; by default you can have up to 10 centered columns.



The **maximum number of columns** in a matrix is determined by the counter `MaxMatrixCols`, default value .. 10, which you can change if necessary using LATEX's `\setcounter{}` or `\addtocounter{}` commands.



If you need left or right alignment in a column or other special formats you may use the `array` environment, or the `mathtools` package which provides starred variants of these environments with an optional argument to specify left or right alignment.

To produce a small matrix suitable for use in text, there is a `smallmatrix` environment, e.g., $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, that comes closer to fitting within a single text line than a normal matrix. Delimiters must be provided. The `mathtools` package provides `p`, `b`, `B`, `v`, and `V` versions of `smallmatrix`, as well as `*` variants as described above. The above example was produced by

```
\bigl( \begin{smallmatrix}
  a&b\\ c&d
\end{smallmatrix} \bigr)
```

`\hdotsfor[] {n}` produces a row of dots in a matrix spanning the given number of columns. For example,

$a \quad b \quad c \quad d$	<code>\begin{matrix} a&b&c&d\\ e&\hdotsfor{3} \end{matrix}</code>	$a \quad b \quad c \quad d$
$e \quad \dots\dots\dots$	<code>\end{matrix}</code>	$e \quad \dots$

Fig. 7 – Theory and practice: the `\hdotsfor{n}` command at work. With (right) and without (left) `colortbl` conflicts.

This construct doesn't work with MATHJAX. Even in LATEX there are reported interferences with `colortbl`, see [stackoverflow entry Interaction between \hdotsfor and colortbl: Bug or Feature?](#) Results of this interference are shown in Figure 7, right and Figure 8, right: the dots just spans a region without the marginal columns of the specified range. It is described as a conflict of `\fill` commands.

If the command works the *spacing* of the dots can be varied through use of a square-bracket option, for example, `\hdotsfor[1.5]{3}`. The number in square brackets will be used as a multiplier, i.e., the normal value is 1.0. Figure 8, left, shows the effect of spacing two. The matrix on the right shows the `colortbl` conflict, and a regular `\dots` solution produced by the code

```
\begin{pmatrix} D_1t&-a_{12}t_2&\dots&-a_{1n}t_n\\
-a_{21}t_1&D_2t&\dots&-a_{2n}t_n\\
\hdotsfor[2]{4} \\ \dots&\dots&\dots&\dots\\
-a_{n1}t_1&-a_{n2}t_2&\dots&D_nt\end{pmatrix}
```

$$\begin{pmatrix} D_1t & -a_{12}t_2 & \dots & -a_{1n}t_n \\ -a_{21}t_1 & D_2t & \dots & -a_{2n}t_n \\ \dots & \dots & \dots & \dots \\ -a_{n1}t_1 & -a_{n2}t_2 & \dots & D_nt \end{pmatrix} (4.1) \begin{pmatrix} D_1t & -a_{12}t_2 & \dots & -a_{1n}t_n \\ -a_{21}t_1 & D_2t & \dots & -a_{2n}t_n \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ -a_{n1}t_1 & -a_{n2}t_2 & \dots & D_nt \end{pmatrix}$$

Fig. 8 – Left: the effect of `\hdotsfor[2]{4}` – Right: `colortbl` conflict in row three, usage of `\dots` in every single field of row four.

5.2 Math spacing commands

The `amsmath` package slightly extends the set of math spacing commands, as shown in Table 1. Both the spelled-out and abbreviated forms of these commands are robust, and they can also be used outside of math.

Table 1 – Space math. The first three examples are made with expressions like `2\ :x\quad 2\negmedspace x`, in the last two rows by joining the left and right macro.

Macro	Example	Macro
<code>\,</code>	<code>\thinspace</code>	<code>2 x 2x</code>
<code>\:</code>	<code>\medspace</code>	<code>2 x 2x</code>
<code>\;</code>	<code>\thickspace</code>	<code>2 x 2x</code>
	<code>2\quad x</code>	<code>2 x y</code>
	<code>2\mspace{18mu}x</code>	<code>2 yx</code>

For the greatest possible control over math spacing, use `\mspace{}` and *math units*. One math unit, or `\mu`, is equal to 1/18 em. Thus to get a negative `\quad{}` you could write `\mspace{-18.0mu}`.

5.3 Dots

For preferred placement of ellipsis dots, raised or on the base line, in various contexts there is no general consensus. It may therefore be considered a matter of taste. By using the semantically oriented commands

- `\dotsc{}` for “dots with commas”
- `\dotsb{}` for “dots with binary operators/relations”
- `\dotsm{}` for “multiplication dots”
- `\dotsi{}` for “dots with integrals”
- `\dotso{}` for “other dots” (none of the above)

instead of `\ldots` and `\cdots`, you make it possible for your document to be adapted 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; see the example below, first the code and then its compilation, where the `\dot` variations are used in the order given by the itemlist above:

```
Then we have the series $A_1, A_2, \dotsc$, the regional
sum $A_1 +A_2 +\dotsb $, the orthogonal product
$A_1 A_2 \dotsm $, and the infinite integral
\[\int_{A_1}\int_{A_2}\dotso.\]
```

Then we have the series A_1, A_2, \dots , the regional sum $A_1 + A_2 + \dots$, the orthogonal product $A_1 A_2 \dots$, and the infinite integral

$$\int_{A_1} \int_{A_2} \dots$$

For most situations, the undifferentiated `\dots` can be used, and `amsmath` will output the most suitable form based on the immediate context; if an inappropriate form results, it can be corrected after examining the output.

5.4 Nonbreaking dashes

A command `\nobreakdash` is provided to suppress the possibility of a line break after the following hyphen or dash. For example, if you write “pages 1– 9” as `pages 1\nobreakdash--9` then a line

break will never occur between the dash and the 9. You can also use `\nobreakdash` to prevent undesirable hyphenations in combinations like $\mathbb{p}\mathbb{9}$ -adic. For frequent use, it's advisable to construct new macros, e.g.,

```
\newcommand{\p}{\mathbb{p}\nobreakdash}% for "\mathbb{p}-adic"
\newcommand{\Ndash}{\nobreakdash--}% for "pages 1\Ndash 9"
% For "\n dimensional" ("n-dimensional"):
\newcommand{\n}[1]{\mathbb{n}\nobreakdash-\hspace{0pt}}
```

The last example shows how to prohibit a line break after the hyphen but allow normal hyphenation in the following word. To accomplish this it suffices to add a zero-width space after the hyphen.

5.5 Accents in math

In ordinary LATEX the placement of the second accent in doubled math accents is often poor. With the `amsmath` package you will get improved placement of the second accent, i.e., `\hat{\hat{A}}` compiles to $\hat{\hat{A}}$ and triggers a customization of the line height.

The commands `\dddote` and `\ddddot` are available to produce triple and quadruple dot accents \ddot{a} and $\overset{\cdot}{\ddot{a}}$ in addition to the `\dot` and `\ddot` accents \dot{a} and \ddot{a} already available in LATEX.

To get a superscripted hat or tilde character, load the `amsxtra` package and use `\sphat` or `\sptilde`. Usage is `A\sphat`. Note the absence of the superscript operator $\hat{\cdot}$.

To place an arbitrary symbol in math accent position, or to get under accents, see the `accents` package by Javier Bezos. And make sure to load `amsmath` before `accents`.

5.6 Roots

Sometimes we might not be happy with the default LATEX placement of root indices, e.g., `\sqrt[\beta]{k}` looks like $\sqrt[\beta]{k}$. In the `amsmath` package `\leftroot{}` and `\uproot{}` allow you to adjust the position of the root:

```
\sqrt[\leftroot{-2}\uproot{2}\beta]{k}
```

will move the beta up and to the right: $\sqrt[\beta]{k}$. The negative argument used with `\leftroot{-2}` moves the β to the right. The units are normalized to a small amount that is a useful size for such adjustments.

5.7 Boxed formulas

The command `\boxed{}` puts a box around its argument, like `\fbox{}` except that the contents are in math mode, e.g.

$$\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))} \quad (11)$$

is the result of

```
\begin{equation}
\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))}
\end{equation}
```

5.8 Over and under arrows

Basic LATEX provides `\overrightarrow{}` and `\overleftarrow{}` commands. Some additional over and under arrow commands are provided by the `amsmath` package to extend to concepts shown in Table 2.

Table 2 – More macros for super and sub arrows. For the underarrows I introduced some p’s and q’s to show the dependence on the decenders of the characters.

arrow	over .. text	under .. math
.. left ..	\overleftarrow{wvec}	\overleftarrow{wvec}
.. right ..	$\overrightarrow{widevector}$	$\overrightarrow{wiveveqtor}$
.. leftright ..	$\overleftrightarrow{wvec}$	$\overleftrightarrow{wvec}$

5.9 Extensible arrows

`\xleftarrow{}` and `\xrightarrow{}` produce arrows that extend automatically to accommodate unusually wide subscripts or superscripts. These commands take one optional subscript argument and

one superscript argument:

$$A \xleftarrow{\overrightarrow{n+\mu-1}} B \xrightarrow[T]{n\pm i-1} C \xleftrightarrow{?} D \xrightarrow[\text{sunrise}]{} E \quad (12)$$

```
A\xleftarrow{\overrightarrow{n+\mu-1}}%
B\xrightarrow[T]{n\pm i-1}%
C\xrightarrow[?]{\overleftarrow{\mspace{36mu}}}%
D\xrightarrow[\text{sunrise}]{}E
```

Here MATHJAX provides only some idea of the concept, while the LATEX compilation seems to include more layout considerations.

5.10 Affixing symbols to other symbols

LATEX provides `\stackrel{}{}` for placing a superscript above a binary relation. In the `amsmath` package there are somewhat more general commands, `\overset{}{}` and `\underset{}{}`, that can be used to place one symbol above or below another symbol, whether it's a relation or something else. The input `\underset{*}{X}` is compiled to X_* while

`\overset{*}{X}` is the analog for adding a symbol above like $\overset{*}{X}$. The command `\overunderset{o}{u}` is a combination of these, taking three arguments to place superscript sized expressions above and below the same base.

In the paragraph above we can see that these under- and overscripts both add some line height. So it's probably not a good idea to use these commands inline.

6 Text in Math environment

The main use of the command `\text{}{}` is for words or phrases in a display. It is very similar to the LATEX command `\mbox{}{}` in its effects, but has a couple of advantages. If you want a word or phrase of text in a subscript, you can type `|\dots\{word or phrase\}|`, which is slightly easier than the `\mbox{}{}` equivalent: `|\dots\{\mbox{\rmfamily\scriptsize word or phrase}\}`. Note that the standard `\textrm{}{}` command will use the `amsmath` `\text{}{}` definition, but ensure the `\rmfamily` font is used.

References

- [1] Leslie Lamport. *LaTeX*. Addison-Wesley, 2nd edition, 1994.
- [2] Frank Mittelbach, Michel Goossens, Johannes L. Braams, David P. Carlisle, and Chris A. Rowley. *The LaTeX Companion 2*. Tools and Techniques for Computer Typesetting. Addison-Wesley Professional, Reading, Massachusetts, second edition, April 2004.