The **eqnlines** Package
Reference Manual

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland

`nbeisert@itp.phys.ethz.ch`

2026/01/28, v0.14

Abstract

eqnlines is a LATEX 2_ε package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of LATEX and the amsmath package with many options for convenient adjustment of the intended layout. In particular, the package adds flexible schemes for numbering, horizontal alignment and semi-automatic punctuation, and it improves upon the horizontal and vertical spacing options. The extensions can be used and adjusted through optional arguments and modifiers to the equation environments as well as global settings.

Contents

1 Introduction

Typesetting mathematical equations is an undisputed strength of TeX. LaTeX improved the overall management of display equations, for instance by providing optional numbering. It also added elementary functionality for multi-line equations with alignment. Some of its deficiencies were addressed by the multi-line equation environments of the package `amsmath` which have become an established standard for these purposes.

The package `eqnlines` builds upon and extends the functionality of the LaTeX and `amsmath` equation environments with some new features as well as convenient options to adjust the layout where needed. The main additions are as follows:

- Equation numbers can be assigned to individual lines (as for `align` and `gather`) or once for the multi-line equation block (as for `multline`). In the former case, a sub-numbering scheme can be applied (as through `subequations`). In the latter case, the position can be assigned to a specific line (first/middle/last/chosen). Moreover, equation numbers can be turned on and off by commands, and they can be triggered by setting a label.

- The vertical spacing above and below single- and multi-line equations of LaTeX and `amsmath` can be somewhat variable, hard to control and even resistive in certain situations. The package implements clearer structures controlling the vertical spacing, including proper dependency on the text line above and ways to adjust the spacing.

- The framework introduces a scheme which semi-automatically inserts punctuation, e.g. '.' or ',', at the end of the following (or every) equation environment. Punctuation can also be inserted at every alignment column or equation line including the possibility to prepend a certain spacing.

- Next to `\[...\]` as an alias for the single-line `equation` environment, the package uses `\<...\>` as an alias multi-line equations.

- The horizontal alignment and indentation of equation lines can be adjusted via a scheme or on a line-by-line basis.

- The alignment marker can be placed before or after the equation signs while maintaining proper spacing to symbols before and after it. This simplifies the construction of continuing equations in an aligned context.

- Equation lines are subject to shrinking of space if the available space does not suffice (analogously to single-line equations).

- Most settings can be controlled via optional arguments and modifiers to the equation environment or via global settings. This includes switching between different types of equation environments, enabling or disabling numbering, adjusting vertical spacing, etc. This feature simplifies the adjustment and fine-tuning of equations towards the intended layout.

- Last but not least, the underlying `amsmath` code, originating from the TeX era and early LaTeX years, has been redesigned with emphasis on clarity, readability, adjustability and maintainability (but at the cost of moderately higher resource consumption and moderately lower efficiency). Nevertheless, it remains original LaTeX 2_ε code without using the `expl3` layer.

The package represents a stand-alone implementation of an equations environment which is largely compatible with the established LaTeX and `amsmath` environments `equation`, `multline`, `gather`, `align` and their variants. Hence, the package can be used instead of `amsmath` with no or minor modifications to the LaTeX sources for single- and multi-line equations. It can also be used alongside `amsmath` including the `mathtools` extensions to make use of the additional maths typesetting features provided by these packages. In the

latter case, the equation environments of LaTeX and `amsmath` are either replaced or left in place while the `eqnlines` environments can be accessed using the alternate name `equations`.

2 Usage

Notice regarding package version v0.14: Please note that this package is still in a development and testing stage. The present version serves as a release candidate for v1.0. Please report any malfunction that you may notice.

To use the `eqnlines` package add the command

$$\usepackage\{eqnlines\}$$

to the preamble of the LaTeX document. To use unrelated features of the `amsmath` package or of the `mathtools` extension, it makes sense to load these packages *before* `eqnlines`.

2.1 Equations Environment

equations (*env.*) **Options.** The environment `equations` accepts a comma-separated list of optional parameters '[*opts*]':

$$\begin\{equations\}\mathit{mod}\,[\mathit{opts}]\,\mathit{mod}\,\_$$
$$\ldots$$
$$\end\{equations\}\,\_$$

Furthermore, the environment accepts modifiers *mod* (like the star modifier '*' for many other LaTeX macros) acting as shortcuts for some options to be explained further below. They can be specified in any order.

We note that the `equations` environment should be started with a whitespace character '␣' which provides a clear separation from optional arguments '[*opts*]' and/or modifiers which must immediately follow the environment declaration `\begin{equations}` without whitespaces. Any character without a proper meaning will also start the equation content, however, future versions of the package may extend the syntax of modifiers, and thus a separation by whitespace is advertised (misplaced modifiers produce warning messages). Similarly, `\end{equations}` allows trailing modifiers and it should be followed by a whitespace to avoid undesired effects or warnings.

\eqnlinesset Most options, but not all, can be set permanently by the macro:

$$\eqnlinesset\{\mathit{opts}\}$$

\eqnaddopt Options for the following equations block are set by the macro:

$$\eqnaddopt\{\mathit{opts}\}$$
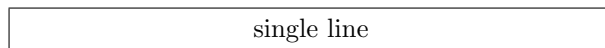
\eqncontrol Several options can be controlled for individual lines or cells within the equations block by the macro:
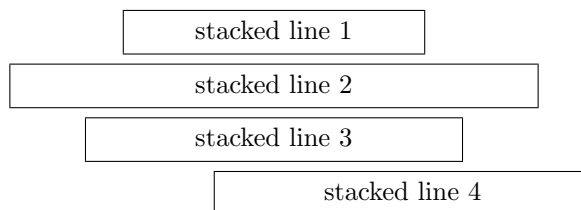
$$\eqncontrol\{\mathit{opts}\}$$

The `\eqncontrol` interface also provides several features for which no other macro definitions exist. Shortcuts to frequently used features could be installed by user definitions such as:

$$\newcommand\{\mathit{shortcut}\}[1]\{\eqncontrol\{\mathit{key}=\{\#1\}\}\}$$
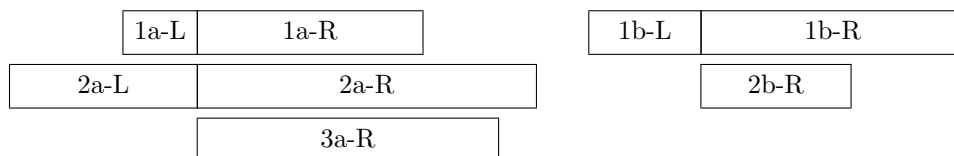
3

Modes of Operation. The package supplies a main maths environment called `equations` which has three principal modes of operation. It can display a single-line equation just as the LaTeX environment `equation` or the symbolic shortcut `\[...\]`:

$$\boxed{\text{single line}}$$

It can display a stack of equations analogous to the `amsmath` environments `gather` and `multline`:[1]

$$
\boxed{\text{stacked line 1}}
$$
$$
\boxed{\text{stacked line 2}}
$$
$$
\boxed{\text{stacked line 3}}
$$
$$
\boxed{\text{stacked line 4}}
$$

It can also display one or several columns of aligned equations analogous to the `amsmath` environment family `align`:

$$
\boxed{\text{1a-L}}\ \boxed{\text{1a-R}} \qquad \boxed{\text{1b-L}}\ \boxed{\text{1b-R}}
$$
$$
\boxed{\text{2a-L}}\ \boxed{\text{2a-R}} \qquad \boxed{\text{2b-R}}
$$
$$
\boxed{\text{3a-R}}
$$

`single` (*key*)
`lines` (*key*)
`columns` (*key*) The three modes of operation are selected by setting an optional argument as follows:

purpose	single-line equation	stacked equation(s)	aligned equations
name	`single`	`lines`	`columns`
alt. names	`equation`, `eq`, `1`	`gather`, `ga`, `ln`	`align`, `al`, `col`
symbolic	`\[␣...\]␣`	`\<=␣...\>␣`	`\<␣...\>␣`
`amsmath` env.	`equation`	`gather`, `multline`	`align`
columns	—	single	multiple, aligned
alignment	adjustable	adjustable	alternating right/left
parsing	single, direct	two passes	two passes
numbering	on/off	off/single/multiple	off/single/multiple

The aligned mode more or less encompasses all three modes, and the stacked mode with only a single line is more or less just a single equation. However, the more complex forms also come along with some restrictions, hence, it makes sense to use the appropriate mode for the intended equation content. For instance, a single equation simply reads the equation input once, while the multi-line equation environments parse the environment body twice which can potentially disrupt some other functionality that is included in the body. Furthermore, the horizontal adjustment options are very restricted in aligned mode, and therefore the aligned form can automatically reduce to the stacked form (with right alignment) if only a single column is provided (no '`&`'s).

```
\begin{equations}[single]
x=\cos\phi
\end{equations}
```
$$x = \cos\phi \tag{1}$$

---

[1] Arguably, a single-line equation is just a stack of equations of height 1. Nevertheless, there is a single-line mode which prohibits line breaks and which works slightly more efficiently: For example, the multi-line modes will process the input twice which is not needed for the single-line mode. Apart from that, the package takes care that the layout and spacing of single-line equations and multi-line equations consisting of a single line is the same.

```
\begin{equations}[lines]
x=\cos\phi \\ \phi=\arccos x
\end{equations}
```
$$x = \cos\phi \qquad (2)$$
$$\phi = \arccos x \qquad (3)$$

```
\begin{equations}[columns]
x&=\cos\phi & \phi&=\arccos x \\
 &=(z+z^{-1})/2 & &=-i\log z
\end{equations}
```
$$x = \cos\phi \qquad \phi = \arccos x \qquad (4)$$
$$= (z + z^{-1})/2 \qquad = -i\log z \qquad (5)$$

\\   As usual, lines are separated by '\\' and columns by '&'. Note that '\\' admits some optional
&   parameters:

$$\backslash\backslash[\tilde{\ }][*][skip]$$

The combination [skip] introduces additional glue between the lines; the modifier '*' inhibits
a page break, and '~' indicates a continued equation which suppresses punctuation (see
section 2.4) and defers numbering (see section 2.2).

\[...\]   **Alternative Forms.**   The package offers several alternative names for the same mode as
\<...\>   well as a symbolic short form \<...\> extending the LaTeX display equation form \[...\]
= (key)   to multi-line equations.   An additional equal sign '=' in \<=␣...\> serves as a modifier
- (key)   character which acts as a short form for the optional argument lines selecting the lines
| (key)   mode.   Similarly, the modifiers minus '-' and bar '|' select single-line and columns mode,
sqropt (key)   respectively. Both short forms can be customised by setting default arguments via the global
angopt (key)   options sqropt={opts} and angopt={opts}. Both default arguments are preset to nonumber
which disables equation numbering, see section 2.2.

```
\[
x=\cos\phi
\]
```
$$x = \cos\phi$$

```
\<=
x=\cos\phi \\ \phi=\arccos x
\>
```
$$x = \cos\phi$$
$$\phi = \arccos x$$

```
\<
x&=\cos\phi & \phi&=\arccos x \\
 &=(z+z^{-1})/2 & &=-i\log z
\>
```
$$x = \cos\phi \qquad \phi = \arccos x$$
$$= (z + z^{-1})/2 \qquad = -i\log z$$

```
\eqnlinesset{sqropt={donumber}}
\[ x=\cos\phi \]
```
$$x = \cos\phi \qquad (6)$$

equation (env.)   The package also supplies or overwrites the amsmath environments equation, gather,
gather (env.)   multline, align and flalign including their starred -at variants (but not the split con-
multline (env.)   struction). It is possible to define further equation environments env with a predefined set
align (env.)   of options opts using:

$$\backslash[\text{re}]\text{newenvironment}\{env\}\{\backslash\text{eqnaddopt}\{opts\}\backslash\text{equations}\}\{\backslash\text{endequations}\}$$

```
\begin{equation}
x=\cos\phi
\end{equation}
```
$$x = \cos\phi \qquad (7)$$

```
\begin{gather}
x=\cos\phi \\ \phi=\arccos x
\end{gather}
```
$$x = \cos\phi \qquad (8)$$
$$\phi = \arccos x \qquad (9)$$

```
\begin{align}
x&=\cos\phi & \phi&=\arccos x \\
 &=(z+z^{-1})/2 & &=-i\log z
\end{align}
```

$$x = \cos\phi \qquad\qquad \phi = \arccos x \qquad (10)$$
$$= (z + z^{-1})/2 \qquad\quad = -i\log z \qquad (11)$$

```
\newenvironment{eqnlist}
  {\eqnaddopt{lines,shape=left}\equations}
  {\endequations}
\begin{eqnlist}[nonumber]
x=\cos\phi \\ \phi=\arccos x
\end{eqnlist}
```

$$x = \cos\phi$$
$$\phi = \arccos x$$

---

transpose *(key)*   **Transposition.** When the aligned mode is used to produce more than one column of
/ *(key)*   equations, the default line-by-line ordering of the content may be inconvenient. The package
offers a transposition mode `transpose=plain` in which the content is specified on a column-
by-column basis. Columns are separated by '`\&`' (the character '&' must be escaped as '`{\&}`'
in this mode) and the lines within each column are broken by '`\\`' as usual. The continued
transposition mode `transpose=cont` (abbreviated by the modifier '`/`') furthermore reduces
the input by assuming that all secondary alignment markers '`&`' indicate a continued equation
and imply a line break with an empty left equation cell. Note that the transposition is
implemented by reprocessing the input, which imposes some restrictions: all line and column
breaks '`\\`', '`\&`' must be explicit (must not be produced by macro expansion), line breaks
should not use optional arguments (they only work on the first column), and each section
separated by '`\&`' should describe only a single column with one alignment marker per line
(unless in continued transposition mode). Furthermore, the continued mode processes the
alignment marker '`&`', which may cause issues when nesting aligned content.

```
\<[transpose=plain]
x &= \cos\phi \\ &= (z+z^{-1})/2
\&
\phi &= \arccos x \\ &= -i\log z
\>
```

$$x = \cos\phi \qquad\qquad \phi = \arccos x$$
$$= (z + z^{-1})/2 \qquad\quad = -i\log z$$

```
\<[transpose=cont]
x &= \cos\phi &= (z+z^{-1})/2
\&
\phi &= \arccos x &= -i\log z
\>
```

$$x = \cos\phi \qquad\qquad \phi = \arccos x$$
$$= (z + z^{-1})/2 \qquad\quad = -i\log z$$

2.2 Numbering

numberline *(key)*   **Numbering Schemes.** The package extends the established interface of LaTeX and the
n *(key)*   amsmath package for labelling equations with numbers or with manually assigned tags. For
multi-line equations, there are two distinct modes of operations: individual labelling of the
equation lines or one overall number/tag for the whole block of equations. The modes
are selected by an optional argument `numberline=`*mode* (alternatively `nline` or just `n`) as
follows:

name	alt.	description	preset
all	a	individual	all lines
sub	s	lines	subequations (a, b, c, . . .)
first	f		first line
last	l		last line
out	o		last/first line for right/left tags
in	i	single line	first/last line for right/left tags
middle	m*		middle line (rounded down/up for right/left tags)
here	h		line indicated by \numberhere
best	+		line with most available space
top	t		at top
bottom	b		at bottom
center	c	between	at vertical centre (single line at baseline)
center!	c!	lines	at vertical centre (also single line)
median	m		middle line (at baseline or between lines)
center*	c*		tag baseline centred between outer baselines
multi	@		individual lines, numbering on
none	−	mode switch	individual lines, numbering off
single	1		previous single-line mode, numbering on
on	!	activation	turn numbering on
off	*		turn numbering off

```
\begin{equations}[!,numberline=...]
  x  &= \cos\phi  \\ &= (z+z^{-1})/2 \\
\phi &= \arccos x \\ &= -i\log z
\end{equations}
```

all:
$$x = \cos\phi \tag{12}$$
$$= (z + z^{-1})/2 \tag{13}$$
$$\phi = \arccos x \tag{14}$$
$$= -i\log z \tag{15}$$

sub:
$$x = \cos\phi \tag{16a}$$
$$= (z + z^{-1})/2 \tag{16b}$$
$$\phi = \arccos x \tag{16c}$$
$$= -i\log z \tag{16d}$$

best:
$$x = \cos\phi \tag{17}$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z$$

first:
$$x = \cos\phi \tag{18}$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z$$

last:
$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z \tag{19}$$

middle:
$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x \tag{20}$$
$$= -i\log z$$

top:
$$1 + \cfrac{1}{1 + \cfrac{1}{1 + \ldots}} \tag{21}$$

bottom:
$$1 + \cfrac{1}{1 + \cfrac{1}{1 + \ldots}} \tag{22}$$

center!:
$$1 + \cfrac{1}{1 + \cfrac{1}{1 + \ldots}} \tag{23}$$

median:
$$x = -\int \sin\phi\,\mathrm{d}\phi$$
$$= \cos\phi \tag{24}$$

center*:
$$x = -\int \sin\phi\,\mathrm{d}\phi$$
$$= \cos\phi \tag{25}$$

center:
$$x = -\int \sin\phi\,\mathrm{d}\phi \tag{26}$$
$$= \cos\phi$$

evadetag (*key*) Note that the mode best (line with most available space) is activated automatically if the (single) tagged line does not have sufficient space to hold the tag. This feature can be controlled by the setting evadetag=*bool*.

\nonumber **Activation and Selection.** Numbering can be turned on and off (for individual lines or \donumber for the block as a whole depending on the mode) by means of:

nonumber (*key*) The numbering can be disabled or enabled for the block by the keys `nonumber` or `donumber`
donumber (*key*) (nn=`*' or dn=`!' for short) or by `number=`*bool* with *bool* either `on` or `off` (among several
number (*key*) alternative forms). Alternatively the number can be switched by using modifiers:
nn,* (*key*)
dn,! (*key*)   \[*␣...\]   and   \[!␣...\]

This allows to define a default behaviour and specify exceptions where they may occur. The star modifier following directly the environment declaration replaces the starred form of environments (`equation*`, etc.) and there is no need to adjust the closing statement.

\numberhere The placement of a single number for an equation block can be adjusted by:
\numbernext
\\~   \numberhere   and   \numbernext

The former macro overrides the position to the present line, the latter macro defers the number to the next line. For example, if an equation is broken into several lines one may use the combination \numbernext \\ to assign the number to the last line. Note that the modified newline '\\~' combines \numbernext and \\

```
\begin{equations}
   x &= \cos\phi \nonumber \\
     &= (z+z^{-1})/2 \\
\phi &= \arccos x \nonumber \\
     &= -i\log z
\end{equations}
```
$$x = \cos\phi$$
$$= (z + z^{-1})/2 \qquad (27)$$
$$\phi = \arccos x$$
$$= -i\log z \qquad (28)$$

```
\begin{equations}*
   x &= \cos\phi \donumber \\
     &= (z+z^{-1})/2 \\
\phi &= \arccos x \donumber \\
     &= -i\log z
\end{equations}
```
$$x = \cos\phi \qquad (29)$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x \qquad (30)$$
$$= -i\log z$$

```
\eqnlinesset{numberline=last}
\<! x &= \cos\phi \\
 \phi &= \arccos x \>
```
$$x = \cos\phi$$
$$\phi = \arccos x \qquad (31)$$

```
\eqnlinesset{angopt=donumber}
\<* x &= \cos\phi \\
 \phi &= \arccos x \>
```
$$x = \cos\phi$$
$$\phi = \arccos x$$

```
\begin{equations}
   x &= \cos\phi \numbernext \\
     &= (z+z^{-1})/2 \\
\phi &= \arccos x \numbernext \\
     &= -i\log z
\end{equations}
```
$$x = \cos\phi$$
$$= (z + z^{-1})/2 \qquad (32)$$
$$\phi = \arccos x$$
$$= -i\log z \qquad (33)$$

```
\eqnlinesset{numberline=here}
\<!
   x &= \cos\phi \\
     &= (z+z^{-1})/2 \\
\phi &= \arccos x \numberhere \\
     &= -i\log z
\>
```
$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x \qquad (34)$$
$$= -i\log z$$

8

```
\eqnlinesset{numberline=first}
\<!
    x &= \cos\phi \numbernext \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \numbernext \\
      &= -i\log z
\>
```

$$
\begin{aligned}
x &= \cos\phi \\
  &= (z + z^{-1})/2 \\
\phi &= \arccos x \\
     &= -i\log z
\end{aligned}
\qquad (35)
$$

\label **Labels and Tags.** Equation numbers can receive LaTeX labels as usual, and they can be
\tag turned into manually assigned tags using the established macros:

$$\texttt{\label}[name]\{label\} \qquad \text{and} \qquad \texttt{\tag}[*][ref]\{tag\}$$

The optional parameter *name* for \label assigns a name to the label which can be referenced
by \nameref. A \tag replaces the equation number, \tag* will drop the decoration by
parentheses. The optional parameter *ref* for \tag defines the representation of references
by \ref.

Note that a label and a tag will always apply to the next number that will be printed, and
only a single label and/or tag may be specified for it. For example, if the present line has
no numbering, but the following line does, \label or \tag will apply to the following line.

The macros \label and \tag can also be instructed to automatically enable number-
ing/tagging for the present line or block via \donumber, see below. By default, number-
ing/tagging is triggered for \tag, but not for \label reflecting the behaviour set forth by
amsmath. By enabling triggering for \label, numbers will be produced only if they have a
chance of being referenced.

label (*key*) The equations environment provides an alternative means to specify labels and tags within
tag (*key*) the optional arguments [*opts*]
labelname (*key*)
taglabel (*key*)   label={*label*},   tag[*]={*tag*},   labelname={*name*},   taglabel={*ref*},

@ (*key*) or via the modifier @{*label*}:

$$\texttt{\[@\{}label\texttt{\}} \ldots \texttt{\]}$$

In particular, in subequations mode (sub), the optional argument label can be used to
assign a label to the parent number addressing the whole equation block.

The above macros may also be used via the keys label, labelname, tag and taglabel of
the interface \eqncontrol.

\eqref The macro \eqref is the standard method for referring to equation numbers via their label.
This method also uses the layout defined below.

$$\texttt{\eqref}\{label\}.$$

\tagform For custom typesetting, \tagform encloses a number/tag with decoration, \tagbox puts the
\tagbox decorated number in a box and \tagboxed combines the two.

\tagboxed The typesetting of equation numbers and tags passes through two macros, one which defines
tagbox (*key*) the layout and another one which adds a decoration by parentheses. These two methods
tagform (*key*) can be adjusted via the options:

$$\texttt{tagbox}[*]=\{code\} \quad \text{and} \quad \texttt{tagform}=\{l\{code\}r\} \quad \text{or} \quad \texttt{tagform*}=\{code\}$$

Here, *code* is some macro code that references the argument '`#1`' containing the number or tag, and l and r can be opening and closing parentheses for the tag presentation.

The above setting may also be changed for individual lines by the corresponding keys of the interface `\eqncontrol`.

```
\eqnlinesset{tagform=[{#1}]}
\eqnlinesset{tagbox={\textcolor{blue}{#1}}}
\<[!,numberline=last]
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z \qquad [36]$$

2.3 Horizontal Placement

layout (*key*)   **Overall Layout.**   First of all, the overall layout can be adjusted between central and left
center (*key*)   alignment via `layout=center`, `layout=left` or `center`, `left` for short.
left (*key*)

```
\<[layout=center]
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z$$

```
\<[layout=left]
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z$$

---

tags (*key*)   Furthermore, numbers and/or tags may be placed on the right or left margin via `tags=right`,
tagsright (*key*)   `tags=left` or `tagsright`, `tagsleft` for short.
tagsleft (*key*)

```
\<[tags=right]!
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

$$x = \cos\phi \qquad (37)$$
$$= (z + z^{-1})/2 \qquad (38)$$
$$\phi = \arccos x \qquad (39)$$
$$= -i\log z \qquad (40)$$

```
\<[tags=left]!
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

$$(41) \qquad x = \cos\phi$$
$$(42) \qquad = (z + z^{-1})/2$$
$$(43) \qquad \phi = \arccos x$$
$$(44) \qquad = -i\log z$$

---

margin (*key*)   **Margins.**   For both layout choices, the margins and line width of an equation block can
marginleft (*key*)   be adjusted by `margin`, `marginleft`, `marginright` or `linewidth`. The equations and corre-
marginright (*key*)   sponding numbers or tags will be fit within these bounds. This feature can be used within
linewidth (*key*)   lists or enumerations to undo an indentation.

```
\[ \indicate{line width} \]
```
line width

```
\[[margin=2em] \indicate{reduced} \]
```
reduced

```
\begin{itemize}
\item first level
  \[ \indicate{default width} \]
  \[[marginleft=0pt]
    \indicate{full width} \]
\end{itemize}
```
• first level

default width

full width

---

tagmargin (*key*)
tagmargin* (*key*)
tagmarginratio (*key*)

In central alignment layout, one can impose a tag margin `tagmargin={`*dimen*`}` which allocates some space to the tag such that equation content is centred in the remaining horizontal space. The margin can also be set to the width of some text by `tagmargin*={`*text*`}` or it can be calculated as the maximum width of tags by `tagmargin` without parameter (default). The option `tagmarginratio={`*ratio*`}` uses the tag margin only for equation blocks with a ratio of tags to rows above the given (decimal) ratio (a value above 1 uses the tag margin only for single equations with tags; default is `0.334`). The option `tagmarginthreshold={`*threshold*`}` uses the tag margin only if the ratio of spacings would be below the given (decimal) threshold (very much off balance; default is `0.5`). The latter two options together with some tag margin can produce a more appealing layout for equation blocks of mixed filling. In the following example, the former two equations are centred on all horizontal space while the latter two equations are centred on the space left of the tag (the ratio of spacings without tag margin would be very small here):

```
\eqnlinesset{tagmarginthreshold=0.7}
\[! \framebox[4em]{} \]
\[! \framebox[8em]{} \]
\[! \framebox[12em]{} \]
\[! \framebox[16em]{} \]
```

(45)
(46)
(47)
(48)

---

leftmargin (*key*)
leftmargin* (*key*)
minleftmargin (*key*)
maxleftmargin (*key*)

In left alignment layout, all equations are left aligned to a left margin (`leftmargin` is initialised to the first level of enumerations and itemisations). It can be set to the width of some text by `leftmargin*={`*text*`}`. Depending on the situation, the left margin may be reduced or extended to `minleftmargin` or `maxleftmargin`, respectively.

```
\eqnlinesset{layout=left}
\<
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

$$x = \cos\phi$$
$$= (z + z^{-1})/2$$
$$\phi = \arccos x$$
$$= -i\log z$$

```
\<[tags=left,!]
    x &= \cos\phi \\
      &= (z+z^{-1})/2 \\
\phi &= \arccos x \\
      &= -i\log z
\>
```

(49) $x = \cos\phi$

(50) $= (z + z^{-1})/2$

(51) $\phi = \arccos x$

(52) $= -i\log z$

fulllength (*key*)	**Column Separation.** The horizontal alignment of columns is fixed for aligned multi-line
mincolsep (*key*)	equations: Each pair of subsequent columns forms a unit which is aligned at the intermediate
maxcolsep (*key*)	alignment marker '&'. These columns are distributed evenly over the available horizontal

space. Here, the outer space left and right of the set of columns is treated on equal footing to the space between the columns (option `fulllength=off`; default), but it can be eliminated so that the outer columns are pushed right to the margin (option `fulllength=on`). A minimum and maximum column separation can be specified via `mincolsep=`*dimen* and `maxcolsep=`*dimen* (defaults are `2em` and `1em`) or the maximum column separation can be disabled by `maxcolsep=off` (which is implied by `fulllength=on`).

```
\<[maxcolsep=2em]
 x &= \cos\phi     & \phi &= \arccos x \\
   &= (z+z^{-1})/2 &      &= -i\log z \>
```

$$x = \cos\phi \qquad \phi = \arccos x$$
$$= (z + z^{-1})/2 \qquad = -i\log z$$

```
\<[maxcolsep=off]
 x &= \cos\phi     & \phi &= \arccos x \\
   &= (z+z^{-1})/2 &      &= -i\log z \>
```

$$x = \cos\phi \qquad\qquad\qquad \phi = \arccos x$$
$$= (z + z^{-1})/2 \qquad\qquad\qquad = -i\log z$$

```
\<[fulllength]
 x &= \cos\phi     & \phi &= \arccos x \\
   &= (z+z^{-1})/2 &      &= -i\log z \>
```

$$x = \cos\phi \hspace{8cm} \phi = \arccos x$$
$$= (z + z^{-1})/2 \hspace{8cm} = -i\log z$$

Alignment Schemes and Control.   For stacks of equations including single equations, there is just a single alignment column whose horizontal alignment can be adjusted via a shape scheme or by manually adjusting individual lines. A shape scheme determines the horizontal alignment for each line and it is specified by the optional argument `shape=`*mode* as follows:

name	alt.	shape	alignment
`default`	`def`	uniform	default
`left`	`l`		left
`center`	`c`	uniform	central
`right`	`r`		right
`first`	`indent, rc`	first/rest	first line indented
`hanging`	`outdent, lc`	first/rest	first line hanging
`steps`	`lcr`	first/intermediate/last	left/centre...centre/right

Note that the `steps` shape comes to use in the replacement `amsmath` environment `multline`.

```
\eqnlinesset{pad=2em}
\<=[shape=...] x = \cos\phi \\ x = (z+z^{-1})/2 \\
   \phi = \arccos x \\ \phi = -i\log z \>
```

left:	center:	right:
$x = \cos\phi$	$x = \cos\phi$	$x = \cos\phi$
$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$
$\phi = \arccos x$	$\phi = \arccos x$	$\phi = \arccos x$
$\phi = -i\log z$	$\phi = -i\log z$	$\phi = -i\log z$

first:	hanging:	steps:
$x = \cos\phi$	$x = \cos\phi$	$x = \cos\phi$
$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$
$\phi = \arccos x$	$\phi = \arccos x$	$\phi = \arccos x$
$\phi = -i\log z$	$\phi = -i\log z$	$\phi = -i\log z$

align *(key)*
shiftto *(key)*
shiftby *(key)*

The alignment preset can be adjusted for individual lines by the controls:

$$\verb|\eqncontrol{align=left|center|right}|$$
$$\verb|\eqncontrol{shiftto|shiftby=|dimen\verb|}|$$

\shoveleft
\shovecenter
\shoveright

or by the macros:

$$\verb|\shoveleft|\verb|\shovecenter|\verb|\shoveright|[*|!][dimen]],$$

indent *(key)*
\shoveby

In contradistinction to amsmath, these macros can be placed anywhere within the cell and they do not take the cell contents as their argument (doing this here will disallow shrinking of glue towards reducing width). The macros accept an optional argument [*dimen*] specifying a variable amount of shift. They also accept the modifiers '*' or '!' for indentation or hanging indentation by the standard indentation amount (indent=2em). Furthermore, \shoveby[*]{*dimen*} shifts the line by the additional amount *dimen* (the star variant shifts to an absolute position relative to the reference position).

padding *(key)*
padleft *(key)*
padright *(key)*

Reference Positions. The reference positions for left, right and central alignment are determined as follows: The central reference position marks the centre of the available horizontal space. The left and right reference positions are given by the ends of the widest line placed centrally. The latter can be adjusted by adding some padding around the widest line via the optional argument padding|padleft|padright[={*dimen*}] while preserving the central default position. The value 'indent' sets the padding to the default indentation amount and 'max' extends the padding to all available space. Note that indent*={*dimen*} sets the default indentation amount and the left padding at the same time.

```
\eqnlinesset{indent=2em,pad=5em}
\<=
\shoveleft   \framebox[5em]{left} \\
\shoveleft*  \framebox[5em]{indent} \\
\shovecenter \framebox[5em]{center} \\
\shoveright  \framebox[5em]{right}
\>
\eqnlinesset{layout=left}
\eqnlinesset{leftmargin=2em}
\eqnlinesset{indent=2em}
\<=
\shoveleft! \framebox[5em]{outdent} \\
\shoveleft  \framebox[5em]{left} \\
\shoveleft* \framebox[5em]{indent} \\
\shoveright \framebox[5em]{right}
\>
```
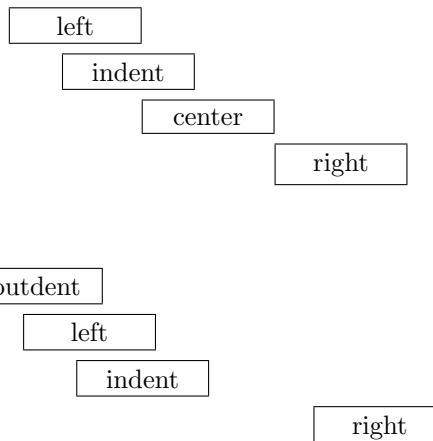
Fitting. Finally, we note that the package will make attempts at fitting the equation components into the horizontal space by adjusting some dimensions with the priority of avoiding overlong lines. The adjustments will first concern the intercolumn and margin

spacing. Secondly, TeX will attempt to shrink the glue between symbols for very wide single and stacked equations (but not aligned equations). Finally, equation tags may be shifted out of the way vertically in order to free up horizontal space. If all attempts fail, overlong lines will be indicated.

alignshrink (*key*) The threshold for shrinking of glue can be controlled by the two parameters `alignshrink`
tagshrink (*key*) and `tagshrink` accepting values ranging between `0` (no shrink) and `4` (full allowable shrink).
alignbadness (*key*) They are used towards determining whether to shift away from the intended alignment
tagbadness (*key*) position or whether to raise or lower the equation tag, respectively. Small values prevent shrinking and higher values allow for more compression. The corresponding parameters `alignbadness` and `tagbadness` accept integer values setting the native threshold in TeX's native units of `\badness`.

`\<=!`	$x + x$ (53)
`x+x \\`	$x + x + x + x$ (54)
`x+x+x+x \\`	$x + x + x + x + x + x$ (55)
`x+x+x+x+x+x \\`	$x + x + x + x + x + x + x + x$ (56)
`x+x+x+x+x+x+x+x \\`	$x+x+x+x+x+x+x+x+x+x$ (57)
`x+x+x+x+x+x+x+x+x+x \\`	$x+x+x+x+x+x+x+x+x+x+x$
`\>`	(58)

---

mintagsep (*key*) If the available space on a line does not suffice to place both the equation and its tag (with a minimum separation of `mintagsep`; default is `0.5em`), a tag will automatically be shifted (lowered or raised depending on whether it is placed on the right or left) to an otherwise
shifttag (*key*) empty line. The `\eqncontrol` control `shifttag=`*dimen* (alternatively `\raisetag*`) may be
`\raisetag*` used to shift a tag up (or down with negative arguments). The control `smashtag=`*dimen*
smashtag (*key*) (alternatively `\raisetag`) may be used to fine-tune the vertical placement when the tag
`\raisetag` requires extra vertical space but some space above or below the tag is unoccupied. It smashes some of the tag's height (or depth with negative arguments) and thus reduces the vertical gap created by the tag. Note that this feature can be used successively with positive
pushtag (*key*) and negative arguments to reduce the space in both directions if available. Where needed,
`\raisetag!` the control `pushtag` (or `\raisetag!`) force-pushes the tag to a separate line and frees up the horizontal space occupied by the tag. The numbering modes `top`, `bottom`, `center`, `median`, `center!` and `center*` are special in that they allow for a continuous vertical placement of
tagbetween (*key*) the tag between two lines. The more flexible placement of tags may also be enabled for the single-lines modes by the option `tagbetween`. Here, both lines must have sufficiently much space available for the tag. If not, the tag is shifted up or down or it is places on separate
tagsnap (*key*) line between the two. The option `tagsnap` defines a range within which the tag baseline snaps to a nearby math baseline.

`\[! \phi = -\int \frac{\mathrm{d}x}{\sqrt{1+x^2}} \]`	$\phi = -\displaystyle\int \frac{\mathrm{d}x}{\sqrt{1+x^2}}$ (59)
`\[! x = \frac{\partial}{\partial \phi}\sin\phi` `\raisetag{0.45\baselineskip} \]`	$x = \dfrac{\partial}{\partial\phi}\sin\phi$ (60)

`\<=![numberline=center] \raisetag*{2pt}`
 `x+x+x+x+x+x+x+x \\`
 `x+x+x+x+x+x+x+x \\`
 `x+x+x+x+x+x+x+x \\`
 `x+x+x+x+x+x+x+x`
 `\>`
	$x + x + x + x + x + x + x + x$
	 $x + x + x + x + x + x + x + x$ (61)
	 $x + x + x + x + x + x + x + x$
	 $x + x + x + x + x + x + x + x$

2.4 Punctuation

Extending proper punctuation across equations is a delicate matter, and maintaining it while redacting the text certainly takes more attention to detail than many authors are willing to afford. A contributing factor is that punctuation marks are harder to spot alongside equation context and somewhat out of place anyway.

`\eqnpunct`
`punct (key)`
The package supplies a semi-automatic scheme by which equations are terminated by a specific punctuation mark.[2] Punctuation marks are set by:

$$\texttt{\eqnlinesset\{punct=\{}punct\texttt{\}\}} \qquad \texttt{\eqnpunct\{}punct\texttt{\}} \qquad \texttt{\[[punct=\{}punct\texttt{\}]} \ldots \texttt{\]}$$

The first form sets and enables a default punctuation mark; the middle form sets the punctuation mark for the next equation environment in line; the final form applies to the equation environment only. For example, one might globally declare 'punct={.}' to terminate all equations with a period '.'. The default behaviour can be adjusted to a comma ',' for an individual equation by declaring '\eqnpunct,' before the equation (i.e. at the end of the textual phrase to which the punctuation mark belongs), at the end of the equation or by using the optional argument [punct={,}]. Likewise, \eqnpunct{}, [punct=~] or [punct={}]

`. , ~ (key)`
eliminates a preset punctuation. The modifiers dot '.', comma ',' and tilde '~' for the equations environment are short forms for using a dot, a comma or disabling punctuation.

`\eqnlinesset{punct=.}`
`The equation`
`\[ x = \cos\phi \eqnpunct{} \]`
`can also be written as`
`\eqnpunct,`
`\[ x = (z+z^{-1})/2 \]`
`where we assume`
`\[ z = \exp(i\phi) \]`
	The equation $$x = \cos\phi$$ can also be written as $$x = (z + z^{-1})/2,$$ where we assume $$z = \exp(i\phi).$$

`\eqnpunctapply`
In situations, where the punctuation must appear before the end of the block, e.g. before a "QED", it can be invoked manually by \eqnpunctapply. Further automatic punctuation is suppressed within the current display.

`punctsep (key)`
For convenience, one may also specify a desired space (or any other code sequence) preceding the punctuation by [punctsep={sep}], e.g. sep=\, or sep=\␣.

`punctcol (key)`
`punctline (key)`
`punctall (key)`
`' (key)`
For multi-line equations, there are two further levels of default punctuation for terminating columns and lines which are specified via the option punctcol and punctline. A punctuation item may also be handed on to the next lower level of punctuation via the starred forms punct* and punctline*. Several levels of punctuation can be specified simultaneously by punctall or via the modifier '`'':

$$\texttt{punctall=\{[[}col\texttt{]}line\texttt{]}main\texttt{\}} \qquad \texttt{\['\{[[}col\texttt{]}line\texttt{]}main\texttt{\}} \ldots \texttt{\]}$$

The special value '~' represents no punctuation and \relax hands down. An empty argument for punctall or '`'' removes all levels of punctuation.

`\<'{,;.}`
`   x &= \cos\phi &`
`\phi &= \arccos x \\`
`   x &= (z+z^{-1})/2 &`
`\phi &= -i\log z \>`
	$x = \cos\phi,$ $\qquad \phi = \arccos x;$
	 $x = (z + z^{-1})/2,$ $\qquad \phi = -i\log z.$

---

[2]Clearly, the implementation of the scheme will take higher efforts than direct coding. Hence, the scheme can be useful in situations where equations typically terminate phrases or where punctuation is otherwise expected in regular patterns.

The package offers means to adjust a standard punctuation (such as `punctall={,,.}`) to
the particular situation:

$$\verb|\\|\mathit{mod}_\sqcup \qquad \verb|\[|\ldots\verb|\]|\,\mathit{mod}_\sqcup \qquad \verb|\<|\ldots\verb|\>|\mathit{mod}_\sqcup \qquad \ldots\verb|\end{equations}|\mathit{mod}_\sqcup$$

The newline command with tilde '`\\~`' describes a break within a continued equation for
which punctuation should be suppressed. Furthermore, the `equations` and `equationsbox`
environments as well as the short forms `\[...\]` and `\<...\>` allow trailing punctuation
modifiers *mod* which place the modified punctuation closer to its natural location than at
the beginning of the block. These can take values:

mod:	~	.	,	'{*punct*}
punctuation:	none	dot	comma	*punct*

Note that the trailing modifiers only work for scanned environments; `equationsbox` and
the single-line `equation` with deactivated `scanequation` do not accept trailing modifiers.
These option can be used to stack several independent continued equations within a block
and to adjust the final punctuation to the flow of the surrounding phrase.

| ```
\eqnlinesset{punctall={,.}}
\< x &= \cos\phi \\~
    &= (z+z^{-1})/2 \\
\phi &= \arccos x \\~
    &= -i\log z \>'{\ldots}
``` | $x = \cos\phi$ <br> $\quad = (z + z^{-1})/2,$ <br> $\phi = \arccos x$ <br> $\quad = -i\log z \ldots$ |

2.5 Math Classes at Alignment

Alignment in multi-line equations breaks equations into components before and after the
alignment position. Unfortunately, this also interrupts TeX's math spacing mechanism which
is based on the math classes assigned to the characters, and there appears to be no direct
way of determining the math class to the previous letter. Therefore, one has to make some
assumptions on the letters that will surround the alignment marker '`&`' in order to obtain
the appropriate spacing also across the alignment.

The `amsmath` environment `align` assumes that the left column ends with an ordinary char-
acter. This leads to the correct spacing when an equation $a = b + c$ is broken before the
equals relation as `a&=b+c`, and also if an equation sequence continues on the next line as
`\\&=d-e`. However, it is difficult to achieve the right spacing if the right-hand side is to be
broken into several lines: For instance, `\\&+f` aligns the subordinate binary operation with
the equals sign (which may be undesirable). Instead placing a phantom equals sign is an
effort that somewhat disrupts source readability.

The package implements a more flexible assignment of math classes at the alignment. The
above default behaviour is invoked by the optional argument `class=ampeq` (`ampeq` for short
or '`<`' as modifier). The optional argument `class=eqamp` (`eqamp` for short or '`>`' as modifier)
imposes math classes at the alignment such that an equation sign should be placed just
before the alignment. Concretely, it inserts `\mathrel{}` classes just before and after the
alignment marker. Furthermore, in case of an empty left alignment cell, the leading math
class is changed to `\mathord{}` so that a following binary operator is not interpreted as a
unary one. For example, the following two expressions produce (almost) identical output:

| ```
\<< a &= b+c \\
    &= d-e \\
    &\mathrel{}\phantom{=} +f \>
``` | $a = b + c$ <br> $\quad = d - e$ <br> $\quad\ + f$ |

```
\<> a =& b+c \\
     =& d-e \\
      & +f \>
```

$$a = b + c$$
$$= d - e$$
$$+ f$$

classout *(key)* Math classes just before and after alignment can be adjusted freely by the optional argu-
classin *(key)* ments:
classlead *(key)*

classout={*class*},   classin={*class*},   classlead={*class*}.

The parameter `classlead` alternatively `classin*` determines the math class just after the
alignment if the cell before alignment is empty. The spacing at the alignment is determined
by the pairing of the last/first character and the selected math class at the alignment:

2.6 Vertical Spacing

Display equations in TeX are considered to be part of the surrounding paragraph of text.
Hence, the vertical spacing depends on the surrounding text, in particular on the width
and depth of the line of text directly preceding the equation. Due to this influence it can
be difficult to manually adjust the spacing accurately. The package adds several options
to control the vertical spacing, and it also implements a uniform behaviour for all types of
equations.

The spacing is determined by combination of several aspects:

Baselines.   First, TeX inserts some glue between lines of text to make them appear as
regular as possible. The amount of inserted glue is determined by TeX's rules which depend
on height, depth and intended baseline separation. This interline spacing also applies to the
lines of displayed equations as well as the interfaces between text and displayed equations.

spread *(key)* The spacing between the lines of a multi-line equation environment can be adjusted via
strut *(key)* `spread={`*dimen*`}` which defaults to `\jot≡3pt` above the normal baseline skip. In addition,
strutdepth *(key)* all equation lines and tags are supplied with struts to ensure a minimum height and depth.
The latter behaviour is controlled by the switch `strut` which takes the values 'on' (default),
'cells', 'tags' or 'off'. The relative depth of such a strut is determined by `strutdepth`
(default 0.3).

While the height/depth of text typically takes rather uniform values, the height/depth of
math content can range wildly with the context (plain equations vs. fractions and matrices).
As displayed equations are normally surrounded by a relatively large amount of glue, it
makes sense to reduce the dependency on the height/depth of math content. Therefore, the
package makes equation environments appear to the surrounding text as a line with a fixed
height and depth, and thus interline glue merely fills some potential gaps of the surrounding
displayheight *(key)* text. The apparent height and depth are defined by `displayheight` and `displaydepth`
displaydepth *(key)* which default to the dimensions of a strut.

Vertical Situation.   Second, the spacing of display equations depends on the width of
the previous line of text. If the math content fits well into the available horizontal space,
the display equation is called short and less glue is needed above the equation. The package
implements this basic TeX feature for all single- and multi-line equation environments.

| ```
example of a long text line:
\[ \mbox{long mode} \]
vs.\ short:
\[ \mbox{short mode} \]
following line
``` | example of a long text line:<br><br>long mode<br><br>vs. short:<br><br>short mode<br><br>following line |

shortmode (*key*) TeX also reduces the amount of glue below short equations (potentially to make their spacing appear more uniform). The package allows to adjust the spacing for short equations via the global option `shortmode=`*mode* where *mode* takes the values:

mode	reduced glue
`off`	disabled
`above`	above short equations (package default)
`belowone`	also below short single-line equations
`belowall`	also below all short multi-line equations

short (*key*)
long (*key*) Short and long amounts of glue can also be enforced for individual equation environments via the optional arguments `short` and `long` taking the values `above`, `below` or `both`.

| ```
example of a long text line:
\[[short] \mbox{forced short} \]
and short:
\[[long] \mbox{forced long} \]
following line
``` | example of a long text line:<br><br>forced short<br><br>and short:<br><br>forced long<br><br>following line |

There are three special situations `cont`, `par` and `top` which trigger different spacings: `cont` describes the situation at the start of an empty horizontal list (invoked by `\noindent`) or when an equation block directly follows another one; here, the space above the equation should be minimal (or even negative to remove the space below the previous equation block). `par` describes the situation at the beginning of a paragraph (invoked by `\par`); here, the space above the equation adds to the space between paragraphs. `top` describes the situation at the top of a vertical list (invoked by `\nointerlineskip`); here, one would typically want no space.

| ```
\hrule\begin{minipage}{\linewidth}
\[ \mbox{top} \]
some text\par
\[ \mbox{par} \]
\[ \mbox{cont} \]
\end{minipage}\hrule
``` | top<br><br>some text<br><br>par<br><br>cont |

Explicit Spacing. Third, the package provides several means to adjust the glue around equations:

noskip (*key*)
medskip (*key*) Next to `short` and `long` the spacing above and below equation environments can be reduced to some other fixed smaller amount via `medskip` or removed altogether via `noskip`. These keys also take the values `above`, `below` or `both`.

`\hrule`
`\[[long] \mbox{long default} \]`
`\hrule`
`\[[medskip] \mbox{medium space} \]`
`\hrule`
`\[[noskip] \mbox{no space} \]`
`\hrule`
	long default
	
	medium space
	
	no space

par (*key*) By default, equation environments end in horizontal mode without indentation. The key `par` controls whether the equation environments end in horizontal mode as usual (value `cont`) or in vertical mode (value `par`, default) with a dedicated amount of glue `belowparskip`. An environment can also be made to end in vertical mode without interline skip (value `top`) using the glue `belowtopskip`. The key `par` can be used for situations when vertical mode is preferred, e.g. for lists following equations where the vertical space above the list is excessive. In the following example, `\hrule` will leave an empty line when not in vertical mode.

`\hrule some text`
`\[ \mbox{cont} \]`
`\hrule some text`
`\[[par] \mbox{par} \]`
`\hrule`
	some text
	cont
	some text
	par

...skip (*key*) Variable amounts of skip can be set via `aboveskip` and `belowskip` or `skip` for both simulta-
\vspace neously. In addition, the package extends the LaTeX `\vspace` mechanism to equation bodies
...space (*key*) where it adds vertical space below the next equation line or below the equation environment. Additional glue can be added above or below equation environments by means of the options `abovespace` and `belowspace`.

Glue Dimensions. The package also maintains several global vertical space settings
...skip (*key*) above*pos*skip and below*pos*skip (sometimes *pos*skip for both):

... *pos*skip	both	description
...long...	`longskip`	regular amount of glue
...short...	–	reduced glue for short equations
...cont...	–	glue when issued from an empty `\noindent` paragraph
...par...	–	glue when starting a paragraph (in vertical mode)
...top...	–	glue when issued at the top of vertical list
...med...	`medskip`	medium amount of glue
...tag...	`tagskip`	minimum glue for outer raised/lowered tags

...mode (*key*) The situations *pos*=cont, `par` and `top` use the respective amount of glue above*pos*skip above the equations and the regular amount of glue `belowlongskip` below. These behaviours may be adjusted by the global options above*pos*mode and below*pos*mode with the values:

value	reduced glue
`long`	regular amount of glue
`short`	reduced glue for short equations
`cont`	amount for empty paragraph
`par`	amount for paragraph (and end the paragraph)
`top`	amount for top (and end the paragraph without interline skip)
`noskip`	no glue
`medskip`	medium amount of glue

Page Breaks.   Finally, the breaking of multi-line equations across pages can be controlled as follows: The setting `allowbreaks` (or `allowdisplaybreaks`) taking values `0` (never) through `4` (permissive) controls the permittivity of page breaks within multi-line equations. The optional arguments `prebreak` and `postbreak` taking values `0` (do not) through `4` (enforce) suggest a break just above or below the equation environment. The command `\displaybreak[`*val*`]` with values `0` through `4` (default) suggests a break below the current line or below the equation environment.

2.7   Further Environments and Features

The package supplies some additional environments and features:

Equation Boxes.   The package provides a boxed equation environment `equationsbox` which can be used within arbitrary math content. It works analogously to `equations` including optional arguments and modifiers, but it offers a reduced range of functionality such as (evidently) no numbering (yet, the `lines` mode accepts multiple columns here). It can also be invoked by the symbolic short form `\<...\>` when called within math mode.

The equations box accepts several arguments: `top`, `center`, `bottom` (or `t`, `c`, `b`) specify the vertical alignment of the box. `margin`, `marginleft`, `marginright` specify additional margin space around the equations box. `colsep` specifies the amount of separation between the columns. `frame[=`*cmd*`]` encloses the equations box by a *cmd* such as `\fbox` which accepts one argument (or a command sequence which ends with a macro accepting one argument). `wrap={{`*cmdl*`}{`*cmdr*`}}` surrounds the equations box by the two commands *cmdl* and *cmdr*.

```
\[ \left\{
\begin{equationsbox}[margin=1em]
    x &= \cos\phi \\
\phi &= \arccos x
\end{equationsbox}
\right\}\]
```
$$\left\{ \begin{array}{l} x = \cos\phi \\ \phi = \arccos x \end{array} \right\}$$

```
$\Longrightarrow \<=[shape=l,frame]
    x = \cos\phi &
\phi = \arccos x \\
    x = (z+z^{-1})/2 &
\phi = -i\log z
\> \Longleftarrow$
```
$$\implies \boxed{\begin{array}{ll} x = \cos\phi & \phi = \arccos x \\ x = (z + z^{-1})/2 & \phi = -i\log z \end{array}} \impliedby$$

Cases and Matrix Environments.   The `cases` environment is reimplemented using the special mode `cases` of `equationsbox` with two left-aligned cells whose separation may be adjusted by the option `condsep`. The equations box is wrapped by a left brace by default; this behaviour may be adjusted by the option `braces` taking the values `off`, `l`, `r`, `lr`. Furthermore, a common intro *text* for all conditions may be added by the option `intro={`*text*`}`; `introtext` sets *text* in text mode with a trailing space. Finally, the option `textcond` determines that right cells are given in mode. All features of `equationsbox`, such as punctuation, are equally available for `cases`.

```
\[ x = \begin{cases}[introtext={if},braces=lr]
1 & \phi = 0 \\
0 & \phi = \pi/2
\end{cases} \]
```
$$x = \left\{\begin{array}{ll} 1 & \text{if } \phi = 0 \\ 0 & \text{if } \phi = \pi/2 \end{array}\right\}$$

```
\['{,.} x = \<[cases,textcond]
\cos\phi & one way \\
(z+z^{-1})/2 & or the other
\eqnpunctapply \> \]
```

$$x = \begin{cases} \cos\phi & \text{one way,} \\ (z + z^{-1})/2 & \text{or the other.} \end{cases}$$

?matrix (*env.*) The set of `matrix` environments are reimplemented using particular combinations of options for `equationsbox`: The matrix is arranged in `lines` mode with centred cells `shape=center`.

spread* (*key*) The line spread is reset by `spread*`. Punctuation is turned off by `punctall={}`. The cells are

style (*key*) rendered in `\textstyle` using the option `style=text`. A small column separation `colsep*` is

colsep* (*key*) selected by `colsep=short`. Finally, the equations box is wrapped by a pair of delimiters l and

delim (*key*) r using the option `delim={lr}`. Again, all features of `equationsbox`, such as punctuation,

matrix (*key*) are equally available for the `matrix` environments. Altogether, the combination is achieved by the shortcut option `matrix={lr}`, where lr may take the following abbreviations:

lr	'.'	r	s	a	c	v	d
brackets	none	round	square	angle	curly	bars	double bars
result	$\ldots$	$(\ldots)$	$[\ldots]$	$\langle\ldots\rangle$	$\{\ldots\}$	$\lvert\ldots\rvert$	$\lVert\ldots\rVert$

```
\[ \begin{pmatrix}
\cos\phi & -\sin\phi \\
\sin\phi & \cos\phi
\end{pmatrix} \]
```

$$\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

```
\[ \<[matrix=v,shape=r]
\cos\phi & -\sin\phi &  0 \\
\sin\phi &  \cos\phi &  0 \\
0        &  0        & -1
\> \]
```

$$\begin{vmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & -1 \end{vmatrix}$$

---

subequations (*env.*) **Collective Numbering.** The environment `subequations` groups equations contained in

subeqtemplate (*key*) the body with a common primary equation number and an extra level of numbering (typically: $a, b, c, \ldots$). The numbering layout can be controlled via `subeqtemplate`. For instance, the default behaviour of adding lowercase latin letters to the parent equation number (`#1`) is achieved by:

$$\texttt{subeqtemplate=\{\#1\textbackslash alph\{\#2\}\}}$$

```
\eqnlinesset
  {subeqtemplate={#1-\roman{#2}}}
\begin{subequations}
\[! x = \cos\phi \]
and
\[! \phi = \arccos x \]
\end{subequations}
```

and

$$x = \cos\phi \tag{62-i}$$

$$\phi = \arccos x \tag{62-ii}$$

---

intertext (*env.*) **Text Intermissions.** The environment `intertext` (equivalently the macro `\intertext`)

\intertext injects a (short) line of text into a multi-line equation while preserving the equation alignment across the text. The `intertext` environment must replace the end-of-line marker '`\\`' between two lines of the equation (to avoid blank lines). The environment accepts several of the vertical spacing adjustments as an optional argument.

```
\< x &= \cos\phi
\intertext[medskip]{and}
\phi &= \arccos x \>
```
and

$$x = \cos\phi$$

$$\phi = \arccos x$$

---

inject (*key*)   **Injection.**   At a lower level, the control `\eqncontrol{inject={`*cmd*`}}` injects some com-
inject* (*key*)   mand sequence *cmd* after the present equation line but before interline spacing. The control
`\eqncontrol{inject*={`*cmd*`}}` injects after interline spacing instead.

```
\< x &= \cos\phi
\eqncontrol{inject=\hrule} \\
\phi &= \arccos x \>
```

$$x = \cos\phi$$
$$\phi = \arccos x$$

---

markline (*key*)   **Line Marks.**   The package provides a mechanism to mark an equation line at the end of
qed (*key*)   the present line or just below. This mechanism can be used to display a QED mark:

$$\eqncontrol\{markline=\{symbol=sym,opts\}$$
$$\eqncontrol\{qed[=\{opts\}]\}$$

The QED symbol may as well be invoked by `\qedhere[`*opts*`]` of amsthm. The starred
variants `markline*`, `qed*` and `\qedhere*` should be used for long lines where the mark
would otherwise smash equation content (equation numbers are avoided automatically).

```
\<[n=l]! x &= \cos\phi
\eqncontrol{markline={symbol=$\sqrt{}$}} \\
\phi &= \arccos x
\eqncontrol{qed={shift=.5ex}} \>
```

$$x = \cos\phi$$
$$\phi = \arccos x$$
$$\sqrt{}$$
$$\text{(63)}$$
$$\text{QED}$$

The options *opts* can be used to adjust the placement by `below` (placed on a separate line
below the present line), `baseline` (smashed at the current baseline), `bottom` (smashed at
the bottom of the present line), to fine-tune the vertical position by `shift=`*dimen* or to
adjust the symbol by `symbol=`*sym*. The default position and symbol can be adjusted by the
global settings `markpos`, `marksymbol` and `qedsymbol`.

\framecell   **Frames.**   The package allows to frame cells of an equation block via issuing a simple
framecell (*key*)   command within the cell:

$$\framecell[cmd] \qquad \text{or} \qquad \eqncontrol\{framecell[=\{cmd\}]\}$$

This command corresponds to `\Aboxed` of mathtools. In particular, when used within
columns or aligned mode, the frame will extend over both right and left alignment com-
ponents of a cell; in order to allocate the right amount of space, it should be issued within
the first cell of the pair. The layout of the frame can be adjusted by the optional argu-
ment *cmd* which defaults to `\fbox`: it must be a macro which accepts one argument (or a
command sequence which ends with a macro accepting one argument). Note: Any semi-
automatic punctuation is included within the frame, see section 2.4. Parts of a cell can be
framed by the amsmath macro `\boxed`, which will not include semi-automatic punctuation.
Furthermore, the height and depth of the box are bounded from below by a strut, see section
2.6.

frametag (*key*)   Similarly, the package allows to frame tags:

$$\eqncontrol\{frametag[=\{cmd\}]\}$$

```\< x &= \cos\phi \\``` ```\framecell \phi &= \arccos x \>```	$x = \cos\phi$ $\boxed{\phi = \arccos x}$
```\[\framecell[\fboxrule2pt\fbox]``` ``` \mbox{important} \eqnpunct! \]```	**important!** (boxed)

```\[! \framecell[\fcolorbox{white}{yellow}]```
``` \eqncontrol{frametag=\fboxsep2pt\fbox}```
``` \mbox{highlight}\]```
	highlight (highlighted) $\boxed{(64)}$

\\ Single-Line Composition. Several short pieces of math content may well fit within a single equation line, typically separated by some amount of space like `\qquad` or `\qquad`.
`\eqnbreak` The package provides several context-aware commands for this purpose: the commands
`\eqnsep` `\eqnsep` and `\eqnbreak` insert some horizontal glue, whereas `\eqnjoin` joins two parts of
`\eqnjoin` an equation with some conjunction. The glue typically includes some amount of shrink in order to accommodate the content when space is sparse. Furthermore, `\eqnsep` and `\eqnbreak` also automatically insert the punctuation for columns and lines, respectively. For convenience, the newline command '`\\`' maps to `\eqnbreak` in single equation mode where it serves no other purpose. The commands have several optional arguments:

$$\texttt{\eqnsep|\eqnbreak|\textbackslash\textbackslash} [\texttt{\~{}}][\texttt{*}|[skip]] \qquad \texttt{\eqnjoin}[\texttt{*}|[skip]]\{conjunction\}$$

The modifier '`~`' suppresses punctuation and '`*`' chooses a shorter glue, whereas the op-
`linesep` (*key*) tional argument *skip* specifies the amount of glue explicitly. The default amounts of glue
`colsep` (*key*) for `\eqnbreak` or '`\\`' and `\eqnsep` are determined by the settings `linesep` and `colsep`, respectively. The starred variants `linesep*` and `colsep*` determine the short amount of glue.

```\[x = \cos\phi \\[4em]``` ```\phi = \arccos x \]```	$x = \cos\phi \qquad\qquad \phi = \arccos x$
```\['{,.} x = (z+z^{-1})/2 \\``` ```\phi = -i \log z \]```	$x = (z + z^{-1})/2, \qquad \phi = -i \log z.$
```\[x = \cos\phi \eqnjoin*{\iff}``` ```\phi = \arccos x \]```	$x = \cos\phi \quad \Longleftrightarrow \quad \phi = \arccos x$

`alt` (*key*) **Alternative Content Description.** The package provides a basic interface to describe
`\eqnalt` the equation content in an alternative form for the purposes of accessibility or documentation (corresponding to the `alt` tag in HTML):

$$\texttt{alt=}\{alt\ text\} \qquad \text{or} \qquad \texttt{\eqnalt}[opt]\{alt\}$$

At the moment the alternative text *alt* is not processed further, but an accessibility extension may implement the feature in tagged PDFs or HTML conversion. The comma-separated optional arguments *opt* may specify the content further: `line` and `cell` restrict the applicability to the current equation line or cell, respectively. Other keys might specify the content format and language.

```\<[alt={example equations}]```
```x &= \cos\phi \\```
```\eqnalt[line]{reverse relationship}```
```\phi &= \arccos x \>```
	$x = \cos\phi$
	
	$\phi = \arccos x$

2.8 General Options

\eqnlinesset Options of general nature can be selected by the commands:

$$\usepackage[\mathit{opts}]\{eqnlines\}$$
$$\text{or}\quad \backslash PassOptionsToPackage\{\mathit{opts}\}\{eqnlines\}$$
$$\text{or}\quad \backslash eqnlinesset\{\mathit{opts}\}$$

`\PassOptionsToPackage` must be used before `\usepackage`; `\eqnlinesset` must be used afterwards. *opts* is a comma-separated list of options.

The package supplies the following general settings:

option	description
defaults=classic	mimic classic LaTeX/amsmath (layout and dimensions)
defaults=eqnlines	eqnlines layout with fontsize-relative dimensions
rescan	rescan environment body for special commands (e.g. `\verb`)
linesfallback	single column in align mode reverts to lines mode
	value `reuse` avoids third measuring pass
ampproof	equip optional argument parsing with protection for '`&`'
equationcr	determine overloading of '`\\`' for single equations
	`off`: native LaTeX error; `error`: package error;
	`break`: insert horizontal glue (see section 2.7)
modifierwarning	invoke a warning for unknown environment modifiers
scanpar	allow scanning of `\par` within equation body
	(e.g., for use in nested `\parbox` or `minipage`)
verbose	track code location in log file for debugging purposes
scanequation	scan single-line environments `equation` and `\[...\]`
native	native processing of single-line equations (reduced functionality)

2.9 Feature Selection and Package Options

The following few settings can only be specified when loading the package, not via `\eqnlinesset`:

option	description
env=none	provide only `equations` and `equationsbox` environments
env=equation	provide/overwrite `equation`, `displaymath` and `\[...\]`
env=amsmath	provide/overwrite `amsmath` environments (including `equation`)
amsmathends=*bool*	patch `amsmath` environments with individual endings
backup=*bool*	backup original `amsmath` environments as `ams...`
ang=*bool*	provide `\<...\>`
eqref=*bool*	provide `\eqref`
matrix=*bool*	provide `amsmath ?matrix` and `cases` environments

If the above settings are explicitly disabled, the package will only supply the general purpose environment `equations` and its boxed cousin `equationsbox`. In that case, the specific equation environments and other features can be activated by the command:

$$\backslash eqnlinesprovide\{\mathit{features}\}$$

features is a comma-separated list of features:

feature	description
env	provide/overwrite environment *env*:
	`equation`, `equation*`, `displaymath`, `subequations`
	`gather`, `multline`, `align`, `flalign`, `xalign`, `xxalign`
	`gather*`, `multline*`, `align*`, `flalign*`, `xalign*`
	`multlined`, `gathered`, `aligned`
env=name	provide environment *env* as *name*
`sqr`	provide `\[...\]`
`ang`	provide `\<...\>`
`eqref`	provide/overwrite macro `eqref`
`tagform`	provide/overwrite macro `\tagform@`
`maketag`	provide/overwrite macro `\maketag@@@`

3 Information

3.1 Copyright

Copyright © 2024–2026 Niklas Beisert

Based on the LaTeX package amsmath: Copyright © 1995, 2000, 2013 American Mathematical Society; 2016–2024 LaTeX Project and American Mathematical Society.

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in `https://www.latex-project.org/lppl.txt` and version 1.3c or later is part of all distributions of LaTeX version 2008 or later.

This work has the LPPL maintenance status 'maintained'.

The Current Maintainer of this work is Niklas Beisert.

This work consists of the files `README.txt`, `eqnlines.ins` and `eqnlines.dtx` as well as the derived files `eqnlines.sty`, `eqnlines.tex`, `eqnlines.pdf`, `eqnlines-src.tex`, `eqnlines-src.pdf`.

3.2 Credits

This package is based on the LaTeX package amsmath (initially named amstex) which in turn is based on the TeX macro system amstex written by Michael Spivak. The initial work of porting amstex to LaTeX was done in 1988–1989 by Frank Mittelbach and Rainer Schöpf. In 1994 David M. Jones added the support for flush-left layout and did extensive improvements to the align family of environments and to the equation number handling in general. Michael Downes at the AMS served as coordinator for the efforts of Mittelbach, Schöpf, and Jones, and has contributed various bug fixes and additional refinements over time. Since 2016, the package has been maintained by the LaTeX Project with contributions by the above and David Carlisle.

This package has been forked from amsmath in accordance with the LPPL, particularly paragraph 6. The original package amsmath is available at CTAN within latex-amsmath. It uses the basic mechanisms for processing numbered multi-line equations as developed in amsmath (environments `equation`, `align`, `gather`, `multline`, `gathered`, `aligned` and related), as well as code implementing these mechanisms. It differs from amsmath in the following aspects:

- The implementations of `split` and methods unrelated to multi-line equations and equation numbering have been dropped.

- Code has been restructured, macros have been renamed and extended.
- Numbering and horizontal adjustment schemes have been unified and extended.
- Options for math classes surrounding the alignment have been added.
- A punctuation scheme has been added.
- Vertical spacing has been redesigned.
- Optional parameters have been added to environments.
- Various configuration options and layout settings have been added.
- Cooperation with hyperref, showkeys and amsmath has been included into the package.

3.3   Files and Installation

The package consists of the files:

`README.txt`	readme file
`eqnlines.ins`	installation file
`eqnlines.dtx`	source file
`eqnlines.sty`	package file
`eqnlines.pdf`	manual

The distribution consists of the files `README.txt`, `eqnlines.ins` and `eqnlines.dtx`.

- Run (pdf)LaTeX on `eqnlines.dtx` to compile the manual `eqnlines.pdf` (this file).
- Run LaTeX on `eqnlines.ins` to create the package `eqnlines.sty`. Copy the file `eqnlines.sty` to an appropriate directory of your LaTeX distribution, e.g. *texmf-root*/`tex/latex/eqnlines`.

3.4   Related CTAN Packages

The package is related to other packages available at CTAN:

- This package uses the package keyval to process the options for the package, environments and macros. Compatibility with the keyval package has been tested with v1.15 (2022/05/29).
- This package reproduces the math environments functionality of the package amsmath. The present code is based on amsmath v2.17t (2024/11/05). Compatibility with the amsmath package is maintained whether eqnlines is loaded before or after amsmath. By default, eqnlines overwrites most math environments of amsmath with its own implementations. It can also preserve them as ams... if needed. Alternatively, eqnlines may assign individual names to the maths environments and preserve the ones of amsmath. The other features provided by amsmath can be used.
- The package mathtools is a popular extension of the amsmath package. This package incorporates some of the features and improvements provided by the mathtools package. Compatibility with the mathtools package has been tested with v1.31 (2024/10/04), and it is maintained whether eqnlines is loaded before or after mathtools. Some features like emphasising equations via empheq do not (yet) work.
- This package has several overlapping goals and considerations with the package breqn in handling long equations more conveniently. However, the approach is very different from breqn which mainly aims at automation.

- This package cooperates with the package hyperref to create anchors and references within the electronic document. Compatibility with the hyperref package has been tested with v7.01l (2024/11/05).

- This package cooperates with the package beamer in assigning default colours for math content. Compatibility with the beamer package has been tested with v3.74 (2025/06/15).

- This package supports the display of labels and references through the package showkeys. Compatibility with the showkeys package has been tested with v3.21 (2024/05/23).

- This package supports placement of QED symbols within proofs through the \qedhere interface of the package amsthm. Compatibility with the amsthm package has been tested with v2.20.6 (2020/05/29).

- This package supports PDF tagging using the framework defined in tagpdf and luamml (when using the lualatex engine). It uses and adjusts some methods from the development framework latex-lab (math and amsmath modules). Compatibility has been tested with the package versions tagpdf v0.99w (2025/10/31), luamml v0.7.0 (2025-10-20) and latex-lab (2025/11/01). Support for PDF tagging is preliminary, partial and fragile; it may break due to future developments of the PDF tagging framework.

- This package is currently not compatible with the package cleveref (thanks to Jonáš Dujava for pointing out). The command \Cref will not refer properly to equation numbers recorded by the equations environment. Further features of either package and/or/in combination with amsmath may fail due to the patching by the package. The alternative package zref-clever appears to work as intended. Incompatibility with the cleveref package has been observed for v0.21.4 (2018/03/27). Compatibility with the zref-clever package has been tested with v0.5.1 (2024/11/28).

3.5 Revision History

v0.14: 2026/01/28

- second release candidate for v1.0
- added cases and matrix environments
- added punctuation modifiers '.' and ',' to \eqnsep, \eqnbreak, \\ and equationsbox
- made \eqnpunctapply work within cells and embedded groups
- made \eqnpunct work for left cells in a column
- added modifiers '<' and '>' for class options ampeq and eqamp
- added option delim for delimiter selection
- added option style for text and display math styles
- added option spread* to reduce line spread

v0.13: 2026/01/14

- release candidate for v1.0
- added trailing modifiers to conveniently adjust punctuation
- added verbose mode for debugging purposes, removed eqnlines-dev package version
- fixed spacing issues with '\\~'

v0.12.2: 2026/01/13

- registered environments with luamml for label processing
- use `\eqnlinesprovide` for starred variants separately

v0.12.1: 2025/12/28

- fixed unwanted deferring of numbering when using `\\~` in continuous mode

v0.12: 2025/12/27

- split off source code documentation into separate document `eqnlines-src`
- added modified newline '`\\~`' to indicate continued equation with deferred numbering and punctuation suppressed
- added `\eqnsep`, `\eqnbreak`, `\eqnjoin` for single-line compositions, mapped '`\\`' for single equations to `\eqnbreak`
- added modifier '`'`' to `equationsbox`
- removed faulty option `crerror`

v0.11: 2025/10/25

- added option `punctall` and modifier '`'`' to specify several levels of punctuation at once

v0.10.1: 2025/06/23

- fix for setting default colours (`math text`) in `beamer`

v0.10: 2025/05/29

- added `numberline` modes `center`, `median`, `top` and `bottom` with continuous vertical adjustments (thanks to Jonáš Dujava for testing)
- fixed spacing following `\paragraph` (thanks to Jonáš Dujava for report)
- added control `inject` to add free-style content after the present line
- added control `markline` and `qed` to display a (QED) mark
- added support for `amsthm` through `\qedhere` (thanks to Jonáš Dujava for suggestion)
- fixed minor issues
- internal structure and minor interface changes

v0.9: 2025/05/18

- option `transpose` to transpose rows and columns in columns mode (thanks to Christophe Bal for suggestion)
- added `\eqncontrol` interface for control within lines and cells
- internal structure and interface changes
- added `\vspace*` for persistent glue at page breaks

- added framed tags (`frametag`)
- added `\raisetag!` to enforce raising (or lowering) of tags even if space is sufficient
- added modifiers, relaxed order, changed lines mode modifier from '`~`' to '`=`'
- fixed minor issues
- thanks to Jonáš Dujava for various reports and suggestions

v0.8: 2025/04/30

- added framed cells (`\framecell`)
- added automatic best line selection for tag placement (`best` and `evadetag`)
- symbolic environment `\<...\>` forwards to `equationsbox` in math mode
- added wrapping for `equationsbox` (`frame`, `wrap`)
- horizontal adjustment reworked and completed; `\shoveby` added
- extended `\label` to assign names to labels for `\namedref`
- interface for alternative representations (`alt` and `\eqnalt`)
- options to adjust line width and margins (`linewidth`, `marginleft`, `marginright`)
- added option `scanpar` to allow `\par` appearing in equation body
- added continuous penalties (`prepenalty`, `postpenalty`, `interpenalty`)
- added overloading for `displaymath` and remaining `amsmath` math environments
- minor interface changes (rename, recombine, values)
- documentation expanded
- several issues fixed

v0.7.1: 2025/04/09

- improvements for PDF tagging
- backup all available math environments at the start using `backup` switch

v0.7: 2025/04/03

- manual expanded, examples added
- fixes for numbering, tagging, options, `linesfallback`, zero lines
- expansions for vertical spacing modes, tag display, `subeqtemplate`
- some consolidations
- internal rearrangements

v0.6.1: 2025/03/27

- `\eqnpunct` can place punctuation within the current equation cell
- `numberline=none` now acts as `numberline=all` and `nonumber`
- fixed and extended `tagmargin` with `tagmarginratio` and `tagmarginthreshold`
- padding now applies to single-line equations as well

v0.6: 2025/03/11

- preliminary PDF tagging support (`https://latex3.github.io/tagging-project/`; amsmath *must* be loaded *before* eqnlines to avoid errors
- classic LaTeX/amsmath vs. eqnlines presets
- changed vertical spacing schemes and added further options
- supplied dimensions processed by `\glueexpr`
- more independent of amsmath structures
- internal reorganisations

v0.5: 2025/02/25

- preview version published on CTAN
- thanks to Till Bargheer for testing and reports