

# The eqnlines Package

Niklas Beisert

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

`nbeisert@itp.phys.ethz.ch`

2025/10/25, v0.11

<https://ctan.org/pkg/eqnlines>

<https://github.com/nbeisert/latex-pkg-nb>

## Abstract

`eqnlines` is a  $\text{\LaTeX} 2_{\epsilon}$  package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of  $\text{\LaTeX}$  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

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>4</b>
2.1	Equations Environment . . . . .	5
2.2	Numbering . . . . .	8
2.3	Horizontal Placement . . . . .	11
2.4	Punctuation . . . . .	16
2.5	Math Classes at Alignment . . . . .	17
2.6	Vertical Spacing . . . . .	18
2.7	Further Environments and Features . . . . .	21
2.8	General Options . . . . .	23
2.9	Feature Selection and Package Options . . . . .	24
<b>3</b>	<b>Information</b>	<b>24</b>
3.1	Copyright . . . . .	24
3.2	Credits . . . . .	25
3.3	Files and Installation . . . . .	25
3.4	Related CTAN Packages . . . . .	26
3.5	Feature Suggestions . . . . .	27
3.6	Revision History . . . . .	27
<b>A</b>	<b>Implementation</b>	<b>29</b>

<b>B</b>	<b>General Support</b>	<b>29</b>
B.1	Development Messages . . . . .	29
B.2	Supporting Definitions . . . . .	29
B.3	Dollardollar Abstraction . . . . .	30
B.4	Look-Ahead in Alignment . . . . .	30
B.5	Error Messages . . . . .	32
B.6	amsmath Integration . . . . .	33
B.7	PDF Tagging Support . . . . .	33
B.8	Key-Value Processing . . . . .	35
<b>C</b>	<b>Parameters and Registers</b>	<b>37</b>
C.1	Parameters . . . . .	37
C.2	Registers . . . . .	40
C.3	Hooks . . . . .	43
<b>D</b>	<b>Features</b>	<b>43</b>
D.1	Punctuation . . . . .	43
D.2	Math Classes at Alignment . . . . .	45
D.3	Framed Cells . . . . .	46
D.4	Alternative Content Description . . . . .	46
<b>E</b>	<b>Equation Numbering</b>	<b>46</b>
E.1	Supporting Definitions . . . . .	46
E.2	Schemes . . . . .	47
E.3	Interface . . . . .	51
E.4	Integration . . . . .	56
E.5	Positioning . . . . .	58
E.6	Component Display . . . . .	62
E.7	Tag Composition . . . . .	65
<b>F</b>	<b>Subequation Numbering</b>	<b>68</b>
F.1	Definitions . . . . .	68
F.2	Environment . . . . .	69
F.3	Subequation Scheme . . . . .	70
<b>G</b>	<b>Display Equations Support</b>	<b>71</b>
G.1	Display Breaks . . . . .	71
G.2	Explicit Vertical Space . . . . .	72
G.3	Default Vertical Spacing . . . . .	74
G.4	Entry and Exit . . . . .	75
G.5	Stack . . . . .	80
<b>H</b>	<b>Multi-Line Support</b>	<b>81</b>
H.1	Measure Support . . . . .	81
H.2	Line Breaks . . . . .	83
H.3	Intertext . . . . .	84
H.4	Line Marks . . . . .	86
<b>I</b>	<b>Column Placement</b>	<b>87</b>
I.1	Supporting Definitions . . . . .	87
I.2	Shape Schemes . . . . .	88
I.3	Width Data . . . . .	91
I.4	Best Line Selection . . . . .	95
I.5	Tag Margin . . . . .	96
I.6	Single Column . . . . .	96

I.7	Multiple Columns	98
<b>J</b>	<b>Single Column Arrangement</b>	<b>103</b>
J.1	Supporting Definitions	103
J.2	Arrangement Methods	104
J.3	Central Alignment	105
J.4	Left Alignment	107
J.5	Right Alignment	108
<b>K</b>	<b>Equations Box Environment</b>	<b>109</b>
K.1	Line Breaks	109
K.2	Stacked Mode	109
K.3	Aligned Mode	111
K.4	Main	112
K.5	Environment	114
<b>L</b>	<b>Single-Line Equation</b>	<b>115</b>
L.1	Native Mode	115
L.2	Print	115
<b>M</b>	<b>Multi-Line with Single Column</b>	<b>116</b>
M.1	Measure	116
M.2	Column Placement	118
M.3	Print	118
<b>N</b>	<b>Multi-Line with Multiple Columns</b>	<b>120</b>
N.1	Support	120
N.2	Transpose	121
N.3	Measure	122
N.4	Columns Placement	125
N.5	Print	125
<b>O</b>	<b>Interface</b>	<b>129</b>
O.1	Scanning the Equation Body	129
O.2	Options Processing	132
O.3	Single-Line Main	134
O.4	Multi-Line Main	135
O.5	Equations Environment	137
<b>P</b>	<b>Options</b>	<b>139</b>
P.1	Selection Tools	139
P.2	Options Declarations	140
P.3	Parameter Presets	151
P.4	Component Selection	153
P.5	Global and Package Options	161

## 1 Introduction

Typesetting mathematical equations is an undisputed strength of  $\text{T}_{\text{E}}\text{X}$ .  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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 2ε` 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 v0.11:** Please note that this package is still in a development and testing stage in the present version. This mainly applies to the documentation

of features and code: Currently, the documentation is basic and minimal without extensive coverage of all features and settings, and it lacks desirable illustrations and examples.

It is likely that some features of the package do not work to full extent, and that the package will not cooperate well with other packages. Therefore, please report any malfunctions that you may notice.

Therefore, it is likely that internal macros and mechanisms will change, It is also conceivable that the public interface will change in minor but relevant ways in order to accommodate for important adjustments or additional features. It is intended that such changes would only require minor adaption of document sources that use an early version of this package.

To use the `eqnlines` package add the command

$$\backslash usepackage{eqnlines}$$

to the preamble of the  $\text{\LaTeX}$  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]`’:

$$\begin{aligned} &\backslash begin{equations} mod [opts] mod\_ \\ &\dots \\ &\backslash end{equations} \end{aligned}$$

Furthermore, the environment accepts modifiers *mod* (like the star modifier ‘`*`’ for many other  $\text{\LaTeX}$  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.

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

$$\backslash eqnlineset{opts}$$

`\eqncontrol` Several options can be controlled for individual lines or cells within the equations block by the macro:

$$\backslash eqncontrol{opts}$$

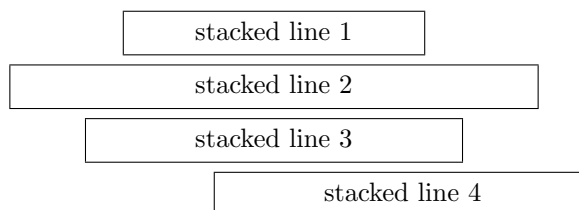
The `\eqncontrol` interface also provides several features for which no other macro definitions exist. Shortcuts to frequently used features could be installed by user definitions such as:

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

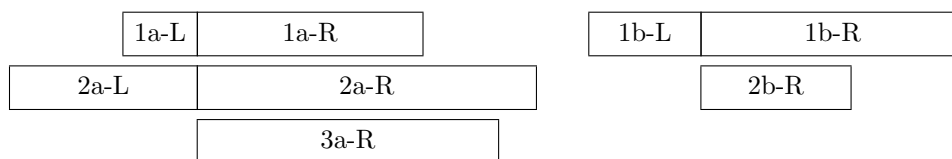
**Modes of Operation.** The package supplies a main maths environment called `equations` which has three principal modes of operation. It can display a single-line equation just as the  $\text{\LaTeX}$  environment `equation` or the symbolic shortcut `\[...\]`:

single line
-------------

It can display a stack of equations analogous to the `amsmath` environments `gather` and `multline`:<sup>1</sup>



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



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

`lines` (*key*)

`columns` (*key*)

purpose	single-line equation	stacked equation(s)	aligned equations
name	<code>single</code>	<code>lines</code>	<code>columns</code>
alt. names	<code>equation</code> , <code>eq</code> , <code>1</code>	<code>gather</code> , <code>ga</code> , <code>ln</code>	<code>align</code> , <code>al</code> , <code>col</code>
symbolic	<code>\[...]</code>	<code>\&lt;=...&gt;</code>	<code>\&lt;...&gt;</code>
<code>amsmath</code> env.	<code>equation</code>	<code>gather</code> , <code>multline</code>	<code>align</code>
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).

---

<code>\begin{equations}[single]</code>			
<code>x=\cos\phi</code>	$x = \cos \phi$		(1)
<code>\end{equations}</code>			
<code>\begin{equations}[lines]</code>			
<code>x=\cos\phi \ \ \ \phi=\arccos x</code>	$x = \cos \phi$		(2)
<code>\end{equations}</code>	$\phi = \arccos x$		(3)
<code>\begin{equations}[columns]</code>			
<code>x&amp;=\cos\phi \ \ \phi&amp;=\arccos x \ \ \</code>	$x = \cos \phi$	$\phi = \arccos x$	(4)
<code>&amp;=(z+z^{-1})/2 \ \ \&amp;=-i\log z</code>	$= (z + z^{-1})/2$	$= -i \log z$	(5)
<code>\end{equations}</code>			

---

<sup>1</sup>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.

`\[...\]` **Alternative Forms.** The package offers several alternative names for the same mode as `\<...\>` well as a symbolic short form `\<...\>` extending the L<sup>A</sup>T<sub>E</sub>X display equation form `\[...\]` to multi-line equations. An additional equal sign ‘=’ in `\<=...\>` serves as a modifier character which acts as a short form for the optional argument `lines` selecting the lines 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.

---

<code>\[</code>			
<code>x=\cos\phi</code>		$x = \cos \phi$	
<code>\]</code>			
<code>\&lt;=</code>			
<code>x=\cos\phi \ \ \ \phi=\arccos x</code>		$x = \cos \phi$	
<code>\&gt;</code>		$\phi = \arccos x$	
<code>\&lt;</code>			
<code>x&amp;=\cos\phi \ \&amp; \ \phi&amp;=\arccos x \ \ \</code>	$x = \cos \phi$	$\phi = \arccos x$	
<code>\&amp;=(z+z^{-1})/2 \ \&amp; \ \&amp;=-i\log z</code>	$= (z + z^{-1})/2$	$= -i \log z$	
<code>\&gt;</code>			
<code>\eqnlineset{sqropt={donumber}}</code>			
<code>\[ x=\cos\phi \]</code>		$x = \cos \phi$	(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` construction). It is possible to define further equation environments `env` with a predefined set of options `opts` using:

`\[re]newenvironment{env}{\eqnadopt{opts}\equations}\endequations`

---

<code>\begin{equation}</code>			
<code>x=\cos\phi</code>		$x = \cos \phi$	(7)
<code>\end{equation}</code>			
<code>\begin{gather}</code>			
<code>x=\cos\phi \ \ \ \phi=\arccos x</code>		$x = \cos \phi$	(8)
<code>\end{gather}</code>		$\phi = \arccos x$	(9)
<code>\begin{align}</code>			
<code>x&amp;=\cos\phi \ \&amp; \ \phi&amp;=\arccos x \ \ \</code>	$x = \cos \phi$	$\phi = \arccos x$	(10)
<code>\&amp;=(z+z^{-1})/2 \ \&amp; \ \&amp;=-i\log z</code>	$= (z + z^{-1})/2$	$= -i \log z$	(11)
<code>\end{align}</code>			
<code>\newenvironment{eqnlist}</code>			
<code>\eqnadopt{lines,shape=left}\equations</code>			
<code>\endequations</code>		$x = \cos \phi$	
<code>\begin{eqnlist}[nonumber]</code>		$\phi = \arccos x$	
<code>x=\cos\phi \ \ \ \phi=\arccos x</code>			
<code>\end{eqnlist}</code>			

---

`transpose` (key) **Transposition.** When the aligned mode is used to produce more than one column of 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           x = \cos \phi           \phi = \arccos x
\&                                           = (z + z^{-1})/2       = -i \log z
\phi &= \arccos x \ \ &= -i\log z
\>

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

---

## 2.2 Numbering

`numberline` (*key*) **Numbering Schemes.** The package extends the established interface of L<sup>A</sup>T<sub>E</sub>X and the `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
<code>all</code>	<code>a</code>	individual	all lines
<code>sub</code>	<code>s</code>	lines	subequations (a, b, c, ...)
<code>first</code>	<code>f</code>	single line	first line
<code>last</code>	<code>l</code>		last line
<code>out</code>	<code>o</code>		last/first line for right/left tags
<code>in</code>	<code>i</code>		first/last line for right/left tags
<code>middle</code>	<code>m*</code>		middle line (rounded down/up for right/left tags)
<code>here</code>	<code>h</code>		line indicated by <code>\numberhere</code>
<code>best</code>	<code>+</code>		line with most available space
<code>top</code>	<code>t</code>	between lines	at top
<code>bottom</code>	<code>b</code>		at bottom
<code>center</code>	<code>c</code>		at vertical centre (single line at baseline)
<code>center!</code>	<code>c!</code>		at vertical centre (also single line)
<code>median</code>	<code>m</code>		middle line (at baseline or between lines)
<code>center*</code>	<code>c*</code>		tag baseline centred between outer baselines
<code>multi</code>	<code>@</code>	mode switch	individual lines, numbering on
<code>none</code>	<code>-</code>		individual lines, numbering off
<code>single</code>	<code>1</code>		previous single-line mode, numbering on
<code>on</code>	<code>!</code>	activation	turn numbering on
<code>off</code>	<code>*</code>		turn numbering off

---



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

<b>all:</b>		<b>sub:</b>		<b>best:</b>	
$x = \cos \phi$	(12)	$x = \cos \phi$	(16a)	$x = \cos \phi$	(17)
$= (z + z^{-1})/2$	(13)	$= (z + z^{-1})/2$	(16b)	$= (z + z^{-1})/2$	
$\phi = \arccos x$	(14)	$\phi = \arccos x$	(16c)	$\phi = \arccos x$	
$= -i \log z$	(15)	$= -i \log z$	(16d)	$= -i \log z$	
<b>first:</b>		<b>last:</b>		<b>middle:</b>	
$x = \cos \phi$	(18)	$x = \cos \phi$		$x = \cos \phi$	
$= (z + z^{-1})/2$		$= (z + z^{-1})/2$		$= (z + z^{-1})/2$	
$\phi = \arccos x$		$\phi = \arccos x$		$\phi = \arccos x$	(20)
$= -i \log z$		$= -i \log z$	(19)	$= -i \log z$	
<b>top:</b>		<b>bottom:</b>		<b>center!:</b>	
$1 + \frac{1}{1 + \frac{1}{1 + \dots}}$	(21)	$1 + \frac{1}{1 + \frac{1}{1 + \dots}}$	(22)	$1 + \frac{1}{1 + \frac{1}{1 + \dots}}$	(23)
<b>median:</b>		<b>center*:</b>		<b>center:</b>	
$x = - \int \sin \phi \, d\phi$		$x = - \int \sin \phi \, d\phi$		$x = - \int \sin \phi \, d\phi$	
$= \cos \phi$	(24)	$= \cos \phi$	(25)	$= \cos \phi$	(26)

**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**      and      **\donumber**

**nonumber** (*key*) The numbering can be disabled or enabled for the block by the keys **nonumber** or **donumber** (**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.

<code>\begin{equations}</code>	
<code>  x &amp;= \cos\phi \nonumber \\</code>	$x = \cos \phi$
<code>  &amp;= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &amp;= \arccos x \nonumber \\</code>	$\phi = \arccos x$
<code>  &amp;= -i\log z</code>	$= -i \log z$
<code>\end{equations}</code>	(28)
<code>\begin{equations}* </code>	
<code>  x &amp;= \cos\phi \donumber \\</code>	$x = \cos \phi$
<code>  &amp;= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &amp;= \arccos x \donumber \\</code>	$\phi = \arccos x$
<code>  &amp;= -i\log z</code>	$= -i \log z$
<code>\end{equations}</code>	(29)
<code>\eqnlineset{numberline=last}</code>	
<code>\&lt;! x &amp;= \cos\phi \\</code>	$x = \cos \phi$
<code>\phi &amp;= \arccos x \&gt;</code>	$\phi = \arccos x$
<code>\eqnlineset{angopt=donumber}</code>	
<code>\&lt;* x &amp;= \cos\phi \\</code>	$x = \cos \phi$
<code>\phi &amp;= \arccos x \&gt;</code>	$\phi = \arccos x$
<code>\begin{equations}</code>	
<code>  x &amp;= \cos\phi \numbernext \\</code>	$x = \cos \phi$
<code>  &amp;= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &amp;= \arccos x \numbernext \\</code>	$\phi = \arccos x$
<code>  &amp;= -i\log z</code>	$= -i \log z$
<code>\end{equations}</code>	(32)
<code>\eqnlineset{numberline=here}</code>	
<code>\&lt;! </code>	$x = \cos \phi$
<code>  x &amp;= \cos\phi \\</code>	$= (z + z^{-1})/2$
<code>  &amp;= (z+z^{-1})/2 \\</code>	$\phi = \arccos x$
<code>\phi &amp;= \arccos x \numberhere \\</code>	$= -i \log z$
<code>  &amp;= -i\log z</code>	(34)
<code>\&gt;</code>	
<code>\eqnlineset{numberline=first}</code>	
<code>\&lt;! </code>	$x = \cos \phi$
<code>  x &amp;= \cos\phi \numbernext \\</code>	$= (z + z^{-1})/2$
<code>  &amp;= (z+z^{-1})/2 \\</code>	$\phi = \arccos x$
<code>\phi &amp;= \arccos x \numbernext \\</code>	$= -i \log z$
<code>  &amp;= -i\log z</code>	(35)
<code>\&gt;</code>	

---

`\label` **Labels and Tags.** Equation numbers can receive L<sup>A</sup>T<sub>E</sub>X labels as usual, and they can be `\tag` turned into manually assigned tags using the established macros:

`\label[name]{label}`      and      `\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 numbering/tagging for the present line or block via `\donumber`, see below. By default, numbering/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}`:

`\[@{label} ... \]`

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.

`\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:

`tagbox[*]={code}` and `tagform={l{code}r}` or `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`.

---

```

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

```

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

[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)`

<pre>\&lt;[layout=center]   x &amp;= \cos\phi \\     &amp;= (z+z^{-1})/2 \\ \phi &amp;= \arccos x \\     &amp;= -i\log z \&gt;</pre>	$x = \cos \phi$ $= (z + z^{-1})/2$ $\phi = \arccos x$ $= -i \log z$
<pre>\&lt;[layout=left]   x &amp;= \cos\phi \\     &amp;= (z+z^{-1})/2 \\ \phi &amp;= \arccos x \\     &amp;= -i\log z \&gt;</pre>	$x = \cos \phi$ $= (z + z^{-1})/2$ $\phi = \arccos x$ $= -i \log z$

**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.

<pre>\&lt;[tags=right]!   x &amp;= \cos\phi \\     &amp;= (z+z^{-1})/2 \\ \phi &amp;= \arccos x \\     &amp;= -i\log z \&gt;</pre>	$x = \cos \phi \tag{37}$ $= (z + z^{-1})/2 \tag{38}$ $\phi = \arccos x \tag{39}$ $= -i \log z \tag{40}$
<pre>\&lt;[tags=left]!   x &amp;= \cos\phi \\     &amp;= (z+z^{-1})/2 \\ \phi &amp;= \arccos x \\     &amp;= -i\log z \&gt;</pre>	$x = \cos \phi \tag{41}$ $= (z + z^{-1})/2 \tag{42}$ $\phi = \arccos x \tag{43}$ $= -i \log z \tag{44}$

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

<pre>\[ \indicate{line width} \]</pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">line width</div>
<pre>\[[margin=2em] \indicate{reduced} \]</pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">reduced</div>
<pre>\begin{itemize} \item first level   \[ \indicate{default width} \]   \[[marginleft=0pt]     \indicate{full width} \] \end{itemize}</pre>	<ul style="list-style-type: none"> <li>• first level</li> </ul> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-top: 10px;">default width</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-top: 10px;">full width</div>

**tagmargin** (*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 **tagmargin\*** (*key*) 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):

---

```

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

```

(45)

(46)

(47)

(48)

---

`leftmargin` (*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.

---

```

\eqnlineset{layout=left}
\<
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 equations: Each pair of subsequent columns forms a unit which is aligned at the intermediate 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 \phi = \arccos x
= (z + z^{-1})/2 \qquad = -i \log z

```

---

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

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

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


---

**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`.

---

<pre>\eqnlineset{pad=2em} \&lt;[shape=...] x = \cos\phi \\ x = (z+z^{-1})/2 \\ \phi = \arccos x \\ \phi = -i\log z \&gt;</pre>		
<b>left:</b>	<b>center:</b>	<b>right:</b>
$x = \cos \phi$	$x = \cos \phi$	$x = \cos \phi$
$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$
$\phi = \arccos x$	$\phi = \arccos x$	$\phi = \arccos x$
$\phi = -i \log z$	$\phi = -i \log z$	$\phi = -i \log z$
<b>first:</b>	<b>hanging:</b>	<b>steps:</b>
$x = \cos \phi$	$x = \cos \phi$	$x = \cos \phi$
$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$
$\phi = \arccos x$	$\phi = \arccos x$	$\phi = \arccos x$
$\phi = -i \log z$	$\phi = -i \log z$	$\phi = -i \log z$

---

`align (key)` The alignment preset can be adjusted for individual lines by the controls:

`shiffto (key)`

`shiftby (key)`

`\eqncontrol{align=left|center|right}`

`\eqncontrol{shiffto|shiftby=dimen}`

`\shoveleft` or by the macros:

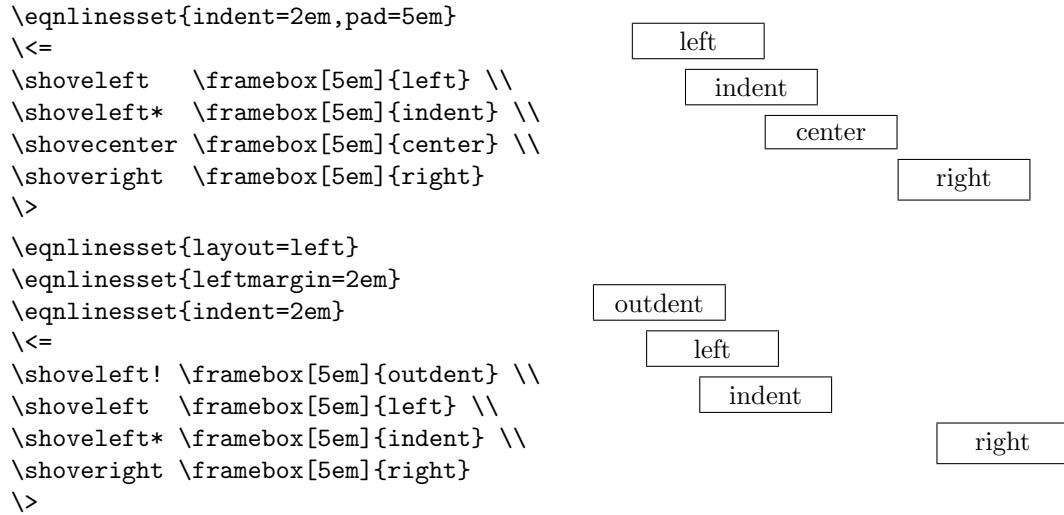
`\shovecenter`

`\shoveright`

`\shoveleft|\shovecenter|\shoveright[*|!|[dimen]],`

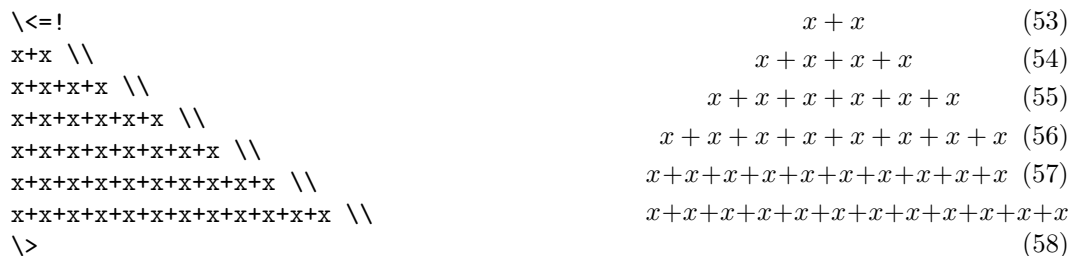
In contradistinction to `amsmath`, these macros can be placed anywhere within the cell and they do not take the cell contents as their argument (doing this here will disallow shrinking of glue towards reducing width). The macros accept an optional argument `[dimen]` specifying a variable amount of shift.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



**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,  $\text{\TeX}$  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** and **tagshrink** accepting values ranging between 0 (no shrink) and 4 (full allowable shrink).  
**tagshrink** (*key*)  
**alignbadness** (*key*) They are used towards determining whether to shift away from the intended alignment 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  $\text{\TeX}$ 's native units of `\badness`.  
**tagbadness** (*key*)



**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 empty line. The `\eqncontrol` control `shifftag=dimen` (alternatively `\raisetag*`) may be used to shift a tag up (or down with negative arguments). The control `smashtag=dimen` (alternatively `\raisetag`) may be used to fine-tune the vertical placement when the tag 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 and negative arguments to reduce the space in both directions if available. Where needed, 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 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 placed on separate line between the two. The option `tagsnap` defines a range within which the tag baseline snaps to a nearby math baseline.

---

$\backslash[! \ \phi = -\int \frac{\mathrm{d}x}{\sqrt{1+x^2}} \ \backslash]$	$\phi = - \int \frac{dx}{\sqrt{1+x^2}}$ <p style="text-align: right;">(59)</p>
$\backslash[! \ x = \frac{\partial}{\partial \phi} \sin \phi \ \backslash]$	$x = \frac{\partial}{\partial \phi} \sin \phi$ <p style="text-align: right;">(60)</p>
$\backslash<=[numberline=center] \ \backslashraisetag*{2pt}$ $\begin{array}{l} x+x+x+x+x+x+x \ \backslash \\ x+x+x+x+x+x+x \ \backslash \\ x+x+x+x+x+x+x \ \backslash \\ x+x+x+x+x+x+x \end{array}$	$\begin{array}{l} x+x+x+x+x+x+x+x+x \\ x+x+x+x+x+x+x+x+x \\ x+x+x+x+x+x+x+x+x \\ x+x+x+x+x+x+x+x+x \end{array}$ <p style="text-align: right;">(61)</p>

---

## 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` The package supplies a semi-automatic scheme by which equations are terminated by a specific punctuation mark.<sup>2</sup> Punctuation marks are set by:

`\eqnlineset{punct={punct}}`    `\eqnpunct{punct}`    `\[[punct={punct}] \dots \]`

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={}]` eliminates a preset punctuation. The modifiers dot `'.'`, comma `'.'` and tilde `'~'` for the

<sup>2</sup>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.



equations environment are short forms for using a dot, a comma or disabling punctuation.

---

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

---

- `\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`.
- `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)` 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
- `punctall (key)` `punct*` and `punctline*`. Several levels of punctuation can be specified simultaneously by `punctall` or via the modifier ‘‘:

`punctall={[[col]line]main}`      `\[‘{[[col]line]main}... \]`

The special value ‘~’ represents no punctuation and `\relax` hands down.

---

<code>\&lt;‘{,;.}</code>		
<code>x &amp;= \cos\phi &amp;</code>	$x = \cos \phi,$	$\phi = \arccos x;$
<code>\phi &amp;= \arccos x \\\</code>		
<code>x &amp;= (z+z^{-1})/2 &amp;</code>	$x = (z + z^{-1})/2,$	$\phi = -i \log z.$
<code>\phi &amp;= -i\log z \&gt;</code>		

---

## 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 character. 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 the readability of the code.

- `class (key)` The package implements a more flexible assignment of math classes at the alignment. The
- `ampeq (key)` above default behaviour is invoked by the optional argument `class=ampeq` (or `ampeq` for
- `eqamp (key)` short). The optional argument `class=eqamp` (or `eqamp` for short) 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:

---

<pre>\&lt;[class=ampeq] a &amp;= b+c \\    &amp;= d-e \\    &amp;\mathrel{=}\phantom{=} +f \&gt;</pre>	$a = b + c$ $= d - e$ $+ f$
<pre>\&lt;[class=eqamp] a =&amp; b+c \\   =&amp; d-e \\   &amp; +f \&gt;</pre>	$a = b + c$ $= d - e$ $+ f$

---

`classout` (*key*) Math classes just before and after alignment can be adjusted freely by the optional arguments:  
`classin` (*key*)  
`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:

		a	<i>a-out</i>		<i>in-b</i>	b		
				<i>lead-c</i>		c		

## 2.6 Vertical Spacing

Display equations in  $\text{\TeX}$  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,  $\text{\TeX}$  inserts some glue between lines of text to make them appear as regular as possible. The amount of inserted glue is determined by  $\text{\TeX}$ '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\equiv 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 text. The apparent height and depth are defined by `displayheight` and `displaydepth` 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 T<sub>E</sub>X feature for all single- and multi-line equation environments.

---

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

---

T<sub>E</sub>X 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:

<i>mode</i>	reduced glue
<b>off</b>	disabled
<b>above</b>	above short equations (package default)
<b>belowone</b>	also below short single-line equations
<b>belowall</b>	also below all short multi-line equations

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

---

	example of a long text line:	
example of a long text line:	<code>\[[short] \mbox{forced short} \]</code>	forced short
and short:		
<code>\[[long] \mbox{forced long} \]</code>	and short:	
following line		forced long
	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.

---

<code>\hrule\begin{minipage}{\linewidth}</code>		top
<code>\[ \mbox{top} \]</code>		
some text\par	some text	
<code>\[ \mbox{par} \]</code>		par
<code>\[ \mbox{cont} \]</code>		
<code>\end{minipage}\hrule</code>		cont

---

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

**noskip** (*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**.

<code>\hrule</code>	
<code>\[[long] \mbox{long default} \]</code>	long default
<code>\hrule</code>	
<code>\[[medskip] \mbox{medium space} \]</code>	medium space
<code>\hrule</code>	
<code>\[[noskip] \mbox{no space} \]</code>	no space
<code>\hrule</code>	

**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.

<code>\hrule some text</code>	some text
<code>\[ \mbox{cont} \]</code>	cont
<code>\hrule some text</code>	
<code>\[[par] \mbox{par} \]</code>	some text
<code>\hrule</code>	par

**...skip** (*key*) Variable amounts of skip can be set via **aboveskip** and **belowskip** or **skip** for both simultaneously. In addition, the package extends the **\vspace** mechanism of L<sup>A</sup>T<sub>E</sub>X to equation bodies 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*) **aboveposskip** and **belowposskip** (sometimes **posskip** for both):

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

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

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

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

## 2.7 Further Environments and Features

The package supplies some additional environments and features:

`equationsbox` (*env.*) **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.

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

`colsep` (*key*)  
`frame` (*key*) `\[ \left\{`  
`wrap` (*key*) `\begin{equationsbox}[margin=1em]`  
`x &= \cos\phi \quad \quad \quad \left\{ \begin{array}{l} x = \cos \phi \\ \phi = \arccos x \end{array} \right\}`  
`\phi &= \arccos x`  
`\end{equationsbox}`  
`\right\}`  
`\Longrightarrow\<=[shape=1,frame]`  
`x = \cos\phi &`  
`\phi = \arccos x \quad \quad \quad \Rightarrow \begin{array}{ll} x = \cos \phi & \phi = \arccos x \\ x = (z+z^{-1})/2 & \phi = -i \log z \end{array} \Leftarrow`  
`x = (z+z-1)/2 &`  
`\phi = -i\log z`  
`\>\Longleftarrow$`

`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, ...). 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:

$$\begin{aligned} & \text{\eqnlineset} \\ & \text{\{subeqtemplate=\{ \#1-\roman\{ \#2 \} \}}} \\ & \text{\begin{subequations}} \end{aligned} \quad x = \cos \phi \quad (62\text{-i}) \\ & \text{\[! x = \cos\phi \]} \quad \text{and} \\ & \text{and} \quad \phi = \arccos x \quad (62\text{-ii}) \\ & \text{\[! \phi = \arccos x \]} \\ & \text{\end{subequations}} \end{aligned}$$

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

$$\begin{array}{l} \backslash < x \&= \cos \phi \\ \backslash \text{eqncontrol}\{\text{inject}=\backslash \text{hrule}\} \backslash \\ \backslash \phi \&= \arccos x \backslash > \end{array} \qquad \frac{x = \cos \phi}{\phi = \arccos x}$$

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

$$\begin{aligned} \text{\<[n=1]! x \&= \cos\phi} & x = \cos \phi & \checkmark \\ \text{\&eqncontrol{markline={symbol=\$\sqrt{\$}}}} & & \\ \text{\phi \&= \arccos x} & \phi = \arccos x & \text{(63)} \\ \text{\&eqncontrol{qed={shift=.5ex}}} & & \text{QED} \end{aligned}$$

22

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

`\framecell[cmd]`      or      `\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 components 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 argument `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}]}`

---

<code>\&lt; x &amp;= \cos\phi \\\</code>	$x = \cos \phi$
<code>\framecell \phi &amp;= \arccos x \&gt;</code>	$\phi = \arccos x$
<code>\[ \framecell[\fboxrule2pt\fbox]</code>	
<code>\mbox{important} \eqnpunct! \]</code>	important!
<code>\[! \framecell[\fcolorbox{white}{yellow}]</code>	
<code>\eqncontrol{frametag=\fboxsep2pt\fbox}</code>	
<code>\mbox{highlight}\]</code>	highlight
	(64)

---

`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):

`alt={alt text}`      or      `\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.

---

<code>\&lt;[alt={example equations}]</code>	
<code>x &amp;= \cos\phi \\\</code>	$x = \cos \phi$
<code>\eqnalt[line]{reverse relationship}</code>	$\phi = \arccos x$
<code>\phi &amp;= \arccos x \&gt;</code>	

---

## 2.8 General Options

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

`\usepackage[opts]{eqnlines}`  
or `\PassOptionsToPackage{opts}{eqnlines}`  
or `\eqnlineset{opts}`

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

The package supplies the following general settings:

option	description
<code>defaults=classic</code>	mimic classic L <sup>A</sup> T <sub>E</sub> X/amsmath (layout and dimensions)
<code>defaults=eqnlines</code>	<code>eqnlines</code> layout with fontsize-relative dimensions
<code>rescan</code>	rescan environment body for special commands (e.g. <code>\verb</code> )
<code>linesfallback</code>	single column in align mode reverts to lines mode value <code>reuse</code> avoids third measuring pass
<code>ampproof</code>	equip optional argument parsing with protection for ‘&’
<code>crerror</code>	invoke an error when ‘\’ is used in a single equation
<code>modifierwarning</code>	invoke a warning for unknown environment modifiers
<code>scanpar</code>	allow scanning of <code>\par</code> within equation body (e.g., for use in nested <code>\parbox</code> or <code>minipage</code> )

## 2.9 Feature Selection and Package Options

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

option	description
<code>env=none</code>	provide only <code>equations</code> and <code>equationsbox</code> environments
<code>env=equation</code>	provide/overwrite <code>equation</code> , <code>displaymath</code> and <code>\[...]</code>
<code>env=amsmath</code>	provide/overwrite <code>amsmath</code> environments (including <code>equation</code> )
<code>amsmathends=bool</code>	patch <code>amsmath</code> environments with individual endings
<code>backup=bool</code>	backup original <code>amsmath</code> environments as <code>ams...</code>
<code>ang=bool</code>	provide <code>\&lt;... \&gt;</code>
<code>eqref=bool</code>	provide <code>\eqref</code>

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:

`\eqnlinesprovide{features}`

*features* is a comma-separated list of features:

feature	description
<i>env</i>	provide/overwrite environment <i>env</i> : <code>equation</code> , <code>gather</code> , <code>multline</code> , <code>align</code> , <code>flalign</code> <code>multlined</code> , <code>gathered</code> , <code>aligned</code> , <code>subequations</code>
<i>env=name</i>	provide environment <i>env</i> as <i>name</i>
<code>sqr</code>	provide <code>\[...]</code>
<code>ang</code>	provide <code>\&lt;... \&gt;</code>
<code>eqref</code>	provide/overwrite macro <code>eqref</code>
<code>tagform</code>	provide/overwrite macro <code>\tagform@</code>
<code>maketag</code>	provide/overwrite macro <code>\maketag@@@</code>

## 3 Information

### 3.1 Copyright

Copyright © 2024–2025 Niklas Beisert



Based on the L<sup>A</sup>T<sub>E</sub>X package `amsmath`: Copyright © 1995, 2000, 2013 American Mathematical Society; 2016–2024 L<sup>A</sup>T<sub>E</sub>X Project and American Mathematical Society.

This work may be distributed and/or modified under the conditions of the L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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` and `eqnlines.pdf`.

## 3.2 Credits

This package is based on the L<sup>A</sup>T<sub>E</sub>X package `amsmath` (initially named `amstex`) which in turn is based on the T<sub>E</sub>X macro system `amstex` written by Michael Spivak. The initial work of porting `amstex` to L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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:

<code>README.txt</code>	readme file
<code>eqnlines.ins</code>	installation file
<code>eqnlines.dtx</code>	source file
<code>eqnlines.sty</code>	package file
<code>eqnlines-dev.sty</code>	package file (development version)
<code>eqnlines.pdf</code>	manual

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

- Run (pdf)L<sup>A</sup>T<sub>E</sub>X on `eqnlines.dtx` to compile the manual `eqnlines.pdf` (this file).
- Run L<sup>A</sup>T<sub>E</sub>X on `eqnlines.ins` to create the package `eqnlines.sty` and the developers version `eqnlines-dev.sty`. Copy the file `eqnlines.sty` to an appropriate directory of your L<sup>A</sup>T<sub>E</sub>X 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 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 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 Feature Suggestions

The following is a list of features for consideration towards future versions of this package. Their potential use may range between useful and niche; and their difficulty between easy and impossible:

- expand documentation further
- complete code documentation
- list of all option keys with scope, defaults and special values

### 3.6 Revision History

**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 L<sup>A</sup>T<sub>E</sub>X/`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

## A Implementation

The appendix documents the various components of the present package.

The code for the package is based on the `amsmath` package, see section 3.1 and section 3.2. It was forked at version v2.17t dated 2024/11/05. Most of the code was substantially redesigned (macros renamed, reshuffled, enhanced), but many of the underlying mechanisms were preserved. The documentation thus contains excerpts from the `amsmath` package documentation explaining some details of the implementation.

Please note that the documentation is completed only for few sections in the present version. Various open issues are remarked.

## B General Support

In the following we describe general purpose supporting routines.

### B.1 Development Messages

The package offers a version `eqnlines-dev` for development and debugging purposes. It outputs extra information on the current location within the code in order to track progress. The extra lines for the development version are indicated as ‘`<dev>`’ in the implementation documentation:

```
1 <dev>\def\eql@dev#1{\PackageInfo{eqnlines-dev}{#1}}
2 <dev>\def\eql@dev@start#1{\eql@dev{starting \string#1}}
3 <dev>\def\eql@dev@enter#1{\eql@dev{entering \string#1}}
4 <dev>\def\eql@dev@leave#1{\eql@dev{ leaving \string#1}}
5 <dev>\def\eql@dev@enterenv{\eql@dev{entering \@currenvir}}
6 <dev>\def\eql@dev@leaveenv{\eql@dev{ leaving \@currenvir}}
7 <dev>\def\eql@dev@in#1#2{\eql@dev{ \space within \string#1 #2}}
```

### B.2 Supporting Definitions

`\eql@false` (*bool*) Rather than the standard L<sup>A</sup>T<sub>E</sub>X scheme of `\xxxfalse`, `\xxxtrue` and `\ifxxx` for boolean variables *xxx*, we use a scheme where `\xxx` is either undefined or defined (to an empty macro) and is tested against by the  $\epsilon$ -T<sub>E</sub>X conditional `\ifdefined\xxx`. In order to make the scheme more tangible, we define the two expected values for boolean variables:

```
8 \let\eql@false\@undefined
9 \let\eql@true\@empty
```

**TODO:** describe

```
10 \def\eql@append#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
11 \def\eql@appendexpand#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
12 \def\eql@appendmacro#1#2{\eql@appendexpand#1{\unexpanded\expandafter{#2}}}
13 \def\eql@letcs#1{\expandafter\let\csname#1\endcsname}
```

### B.3 Dollardollar Abstraction

`\dollar@begin` As of 2025 L<sup>A</sup>T<sub>E</sub>X defines `\dollar@begin` and `\dollar@end` to represent (and adjust) the beginning and end of bare T<sub>E</sub>X display equations (`‘$$’`). For the time being, we make sure to revert to `‘$$’` if these macros are not yet available:

```
14 \ifdefined\dollar@begin
15   \def\eqldollar@begin{\dollar@begin}
16   \def\eqldollar@end{\dollar@end}
17 \else
18   \def\eqldollar@begin{$$}
19   \def\eqldollar@end{$$}
20 \fi
```

### B.4 Look-Ahead in Alignment

Scanning for optional arguments [...] or modifiers such as `‘*’` using the L<sup>A</sup>T<sub>E</sub>X `\ifnextchar` mechanism has two challenges within aligned equations: a square bracket or star may well be part of the intended mathematical expression and the look-ahead could trip upon an alignment character `‘&’` which inadvertently triggers to enter the next alignment column.

`\ifnextchar@loose` To address the first challenge, we can force the special characters to follow immediately the macro invocation. For clarity, we copy L<sup>A</sup>T<sub>E</sub>X’s original `\ifnextchar` in `\kernel@ifnextchar` which skips over spaces as `\eq@ifnextchar@loose`. We replicate the `amsgen` version `\new@ifnextchar` that does not skip over spaces as `\eq@ifnextchar@loose`. The space before `#1` allows to look-ahead for spaces as well:

```
21 \let\eq@ifnextchar@loose\kernel@ifnextchar
22 \long\def\eq@ifnextchar@tight#1#2#3{%
23   \let\reserved@d= #1%
24   \def\reserved@a{#2}%
25   \def\reserved@b{#3}%
26   \futurelet\@let@token\eq@ifnch@tight
27 }
28 \def\eq@ifnch@tight{%
29   \ifx\@let@token\reserved@d
30     \let\reserved@b\reserved@a
31   \fi
32   \reserved@b
33 }
```

`\eq@atxi` Capture `‘@’` as a character (catcode 12) rather than a letter (catcode 11) as `\eq@atxii` so that we can look-ahead for `‘@’` with both `\makeatother` and `\makeatletter` modes:

```
34 \let\eq@atxi=@
35 \begingroup
36   \makeatother
37   \let\tmp=@%
38   \makeatletter
39   \global\let\eq@atxii\tmp
40 \endgroup
```

`\eq@ifnextgobble@...` We introduce a collection of look-ahead macros which do or do not skip over spaces. The macros `\eq@ifstar@...` and `\eq@testopt@...` replicate the L<sup>A</sup>T<sub>E</sub>X counterparts `\ifstar` and `\testopt`. The macros `\eq@ifnextgobble@...` work like `\ifnextchar`, but also gobble the specific character if found; one might define `\eq@ifstar@...` as

`\eq@ifnextgobble@...*`. The macros `\eq@teststaropt@...` tests for combinations of ‘\*’ and optional arguments [...]:

```

41 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
42 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
43 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
44 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
45 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{%
46   \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
47 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{%
48   \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}}
49 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[{\#1}{\#1[{\#2}]}]}
50 \long\def\eq@testopt@tight#1#2{\eq@ifnextchar@tight[{\#1}{\#1[{\#2}]}]}
51 \long\def\eq@teststaropt@loose#1#2#3{%
52   \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}}
53 \long\def\eq@teststaropt@tight#1#2#3{%
54   \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}}
55 \long\def\eq@teststaroropt@loose#1#2#3{%
56   \eq@ifstar@loose{#1}{\eq@testopt@loose{#2}{#3}}}
57 \long\def\eq@teststaroropt@tight#1#2#3{%
58   \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}}
59 \long\def\eq@gobbleopt[#1]{}
60 \long\def\eq@gobbleoptone[#1]#2{}

```

**TODO:** describe

```
61 \def\eq@testopt@default{\eq@testopt@default}
```

**TODO:** describe

```

62 \def\eq@parseopt#1#2{%
63   \def\eq@parseopt@case{#1}%
64   \def\eq@parseopt@end{#2}%
65   \eq@parseopt@peek
66 }
67 \def\eq@parseopt@peek{%
68   \futurelet\eq@parseopt@token\eq@parseopt@select
69 }
70 \def\eq@parseopt@select{%
71   \let\eq@parseopt@next\eq@parseopt@other
72   \ifx\eq@parseopt@token@sptoken
73     \let\eq@parseopt@next\eq@parseopt@end
74   \fi
75   \eq@parseopt@case
76   \eq@parseopt@next
77 }
78 \def\eq@parseopt@other{\eq@parseopt@warn\eq@parseopt@end}
79 \let\eq@parseopt@warn\@empty
80 \def\eq@parseopt@gobble#1{\eq@parseopt@peek}

```

`\eq@spbgrou` The second challenge is addressed by enclosing the look-ahead in spurious groups<sup>3</sup> which  
`\eq@spegrou` protect against triggering ‘&’. The macros `\eq@spbgrou` and `\eq@spegrou` open and  
`\eq@srbgrou` close a spurious group. For some reason, the look-ahead mechanism requires further  
`\eq@sregrou` protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end.  
 These adjustments are included in the macros `\eq@srbgrou` and `\ers@spegrou`:

---

<sup>3</sup>See <https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3040>,  
<https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=amslatex/1834> and  
<https://tex.stackexchange.com/questions/9897/showcase-of-brace-tricks-egroup-iffalse-fi-etc>.

```

81 \def\eql@spbgroup{\iffalse{\fi\ifnum0='}\fi}
82 \def\eql@speggroup{\ifnum0='{ \fi\iffalse}\fi}
83 \def\eql@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
84 \def\eql@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}

```

`\eql@ampprotect` The macros `\eql@ampprotect` and `\eql@ampprotecttwo` inject the opening and closing of `\eql@ampprotecttwo` spurious groups into the look-ahead mechanism:

```

85 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1\eql@sregroup#2}}
86 \long\def\eql@ampprotecttwo#1#2#3{%
87   \eql@srbgroup#1\eql@sregroup#2}{\eql@sregroup#3}}

```

`...@ampsafe` We introduce a collection of ‘&’-safe look-ahead macros:

```

88 \def\eql@ifnextchar@loose@ampsafe#1{%
89   \eql@ampprotecttwo{\eql@ifnextchar@loose#1}}
90 \def\eql@ifnextchar@tight@ampsafe#1{%
91   \eql@ampprotecttwo{\eql@ifnextchar@tight#1}}
92 \def\eql@ifstar@loose@ampsafe{\eql@ampprotecttwo\eql@ifstar@loose}
93 \def\eql@ifstar@tight@ampsafe{\eql@ampprotecttwo\eql@ifstar@tight}
94 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
95 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
96 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
97 \long\def\eql@teststaropt@tight@ampsafe{%
98   \eql@ampprotecttwo\eql@teststaropt@tight}

```

`\eql@amproof` We may want to replace L<sup>A</sup>T<sub>E</sub>X’s definitions `\@ifnextchar`, `\@ifstar` and `\@testopt` to respect ‘&’ characters within aligned equations. This might make unrelated definitions with optional arguments and starred variants more robust in this context. The macro `\eql@amproververt` overwrites the original definitions, and `\eql@amproververt` reverts the changes:

```

99 \let\eql@ifnextchar@org\@ifnextchar
100 \let\eql@ifstar@org\@ifstar
101 \let\eql@testopt@org\@testopt
102 \def\eql@amproververt{%
103   \let\@ifnextchar\eql@ifnextchar@org
104   \let\@testopt\eql@testopt@org
105   \let\@ifstar\eql@ifstar@org
106 }
107 \def\eql@amproof{%
108   \let\@ifnextchar\eql@ifnextchar@loose@ampsafe
109   \let\@testopt\eql@testopt@loose@ampsafe
110   \let\@ifstar\eql@ifstar@loose@ampsafe
111 }

```

## B.5 Error Messages

`\eql@error` Main error and warning message function for the package:  
`\eql@warning`

```

112 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
113 \def\eql@warning{\PackageWarning{eqnlines}}

```

`\eql@error@mathmode` Error messages concerning math mode:

```

114 \def\eql@warn@here#1{\eql@warning{\string#1 not allowed outside equations}}
115 \def\eql@error@mathmode#1{\eql@error{#1 allowed only in paragraph mode}}

```



Warning messages concerning unused and multiply declared labels and tags:

```

\eq@warn@label@unused 116 \def\eq@warn@tags@unused#1#2{\eq@warning{Unused equation #1:
\warn@label@multiple 117   #2 will be lost}}
\eq@warn@tag@unused 118 \def\eq@warn@tags@multiple#1#2#3{\eq@warning{Multiple equation #1:
\eq@warn@tag@multiple 119   previous #2 will be lost#3}}
\eq@warn@name@unused 120 \def\eq@warn@label@unused{\eq@warn@tags@unused{\string\label}
\warn@name@multiple 121   {label '\eq@tags@label'}}
\eq@warn@ref@unused 122 \def\eq@warn@label@multiple#1{\eq@warn@tags@multiple{\string\label's}
\eq@warn@ref@multiple 123   {label '\eq@tags@label'}}{ and replaced by '#1'}}
124 \def\eq@warn@name@unused{\eq@warn@tags@unused{label name}
125   {name declaration}}
126 \def\eq@warn@name@multiple{\eq@warn@tags@multiple{label names}
127   {name declaration}}{}
128 \def\eq@warn@tag@unused{\eq@warn@tags@unused{\string>tag}
129   {tag declaration}}
130 \def\eq@warn@tag@multiple{\eq@warn@tags@multiple{\string>tag's}
131   {tag declaration will be lost}}{}
132 \def\eq@warn@ref@unused{\eq@warn@tags@unused{tag label}
133   {tag label declaration}}
134 \def\eq@warn@ref@multiple{\eq@warn@tags@multiple{tag labels}
135   {tag label declaration}}{}

136 \def\eq@warn@parseopt{%
137   \eq@warning{Unknown modifier token: starting math content}}
138 \def\eq@warn@parseopt@verbose{%
139   \eq@warning{Unknown modifier token: \meaning\eq@parseopt@token}}

```

## B.6 amsmath Integration

We need to overwrite certain macros from amsmath. The method `\eq@amsmath@after` executes argument #1 after loading amsmath is loaded. It also runs the code if amsmath has already been loaded. Furthermore, loading amsmath requires certain macros to be undefined. To this end `\eq@amsmath@before` will execute argument #1 before any future loading of amsmath. `\eq@amsmath@undefine` undefines a macro in this way and `\eq@amsmath@let` overwrites a macro of `\amsmath/`:

```

140 \def\eq@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
141 \def\eq@amsmath@before#1{%
142   \@ifpackageloaded{amsmath}{}{\AddToHook{package/amsmath/before}{#1}}
143 \def\eq@amsmath@undefine#1{\eq@amsmath@before{\let#1\undefined}}
144 \def\eq@amsmath@let#1#2{\eq@amsmath@undefine#1\let#1#2}

```

**TODO:** temporary fix for development stages

```

145 \@ifpackageloaded{amsmath}{}{
146   \DeclareHookRule{package/amsmath/after}
147   {eqnlines}{after}{latex-lab-testphase-math}}

```

## B.7 PDF Tagging Support

`\eq@tagging@...` Proper PDF tagging<sup>4</sup> support requires a L<sup>A</sup>T<sub>E</sub>X (development) version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with L<sup>A</sup>T<sub>E</sub>X versions around 2020: **TODO:** adjust to further developments

```

148 \let\eq@tagging@on\eq@false

```

<sup>4</sup>see <https://latex3.github.io/tagging-project/>

```

149 \IfFormatAtLeastTF{2025-06-01}{%
150   \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}{}
151 \ifdefined\eql@tagging@on
152   \def\eql@tagging@mathsave{%
153     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
154   \def\eql@tagging@mathaddlast{%
155     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
156   \def\eql@tagging@tagbegin{%
157     \UseTaggingSocket{math/display/tag/begin}}
158   \def\eql@tagging@tagend{%
159     \UseTaggingSocket{math/display/tag/end}}
160   \def\eql@tagging@tagsave{%
161     \UseTaggingSocket{math/luamml/mtable/tag/save}}
162   \def\eql@tagging@tagaddbox{%
163     \setbox\z@\copy\eql@tagbox%
164     \UseTaggingSocket{math/luamml/mtable/tag/set}}
165   \def\eql@tagging@tablesaveinner{%
166     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
167   \def\eql@tagging@tableaddinner{%
168     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
169   \def\eql@tagging@tablesavelines{%
170     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
171   \def\eql@tagging@tablesavealign{%
172     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
173   \def\eql@tagging@alignleft{%
174     \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
175   \def\eql@tagging@aligncenter{%
176     \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
177   \def\eql@tagging@alignright{%
178     \UseTaggingSocket{math/luamml/mtable/aligncol}{right}}

```

We need to get hold of the equation body in all cases so that we can feed it into the tagging mechanism:

```

179 \let\eql@single@doscan\eql@true
180 \let\eql@scan@body\eql@scan@body@rescan

```

`\eql@tagging@start` We need to activate tagging for display equations for environments and for enclosures  
`\eql@tagging@end` `[...]` and `<...>`. The tagging interface registration macro `\RegisterMathEnvironment` will work only partially for our cases, hence we replicate code from `\math_register_halign_env:nn`. Make sure collection is not yet active (`\l__math_collected_bool`). Then feed collected environment name, options and body into `\__math_process:nn`. Indicate the start of a display equation:

```

181 \def\eql@tagging@start{%
182   \csname bool_if:N\expandafter\endcsname
183   \csname l__math_collected_bool\endcsname{%
184     \edef\eql@tmp{\@currentvir}{\unexpanded\expandafter{\eql@tagging@opt}}%
185     \the\eql@scan@reg}%
186   \csname __math_process:nn\expandafter\endcsname\eql@tmp
187   \@kernel@math@registered@begin
188   \csname bool_set_true:N\expandafter\endcsname
189   \csname l__math_collected_bool\endcsname
190   }%
191 }
192 \def\eql@tagging@end{}
193 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}
194 \else
195 \def\eql@tagging@mathsave{}

```

```

196 \def\eql@tagging@mathaddlast{}
197 \def\eql@tagging@tagbegin{}
198 \def\eql@tagging@tagend{}
199 \def\eql@tagging@tagsave{}
200 \def\eql@tagging@tagaddbox{}
201 \def\eql@tagging@tablesaveinner{}
202 \def\eql@tagging@tableaddinner{}
203 \def\eql@tagging@tablesavealign{}
204 \def\eql@tagging@tablesavealign{}
205 \def\eql@tagging@alignleft{}
206 \def\eql@tagging@aligncenter{}
207 \def\eql@tagging@alignright{}
208 \def\eql@tagging@start{}
209 \def\eql@tagging@end{}
210 \def\eql@tagging@register@env{\@gobble}
211 \fi

```

## B.8 Key-Value Processing

The package uses the `keyval` mechanism to parse key-value pairs to specify adjustments to the behaviour of the equations environments:

```
212 \RequirePackage{keyval}
```

### Value Selection.

`\eql@decide@select` Some parameter values take values in a given set, e.g. `true` vs. `false` or `left` vs. `right`. The macro `\eql@decide@select` is a general purpose selector. Arguments `#1` and `#2` describe the category and key which are used only towards error messages. Argument `#3` contains the value and argument `#4` is a list of values and corresponding actions in the format

$$\{\{val1a, val1b, \dots\}act1\}, \{\{val2a, val2b, \dots\}act2\}, \dots\}.$$

The (single) value `\relax` matches everything (can be used for handling generic values after specific ones). If no corresponding value is found in the list, an error message is invoked. Single expansion is applied to the list of values:

```

213 \def\eql@decide@relax{\eql@tmpb:=\relax}
214 \def\eql@decide@select#1#2#3#4{%
215   \def\eql@tmpa{#3}%
216   \let\eql@tmpd\@undefined
217   \@for\eql@tmpc:=#4\do{%
218     \ifdefined\eql@tmpd\else
219       \edef\eql@tmpb{\noexpand\eql@tmpb:=\expandafter\@firstoftwo\eql@tmpc}%
220       \ifx\eql@tmpb\eql@decide@relax
221         \def\eql@tmpa{\relax}%
222       \fi
223       \expandafter\@for\eql@tmpb\do{%
224         \ifx\eql@tmpa\eql@tmpb
225           \edef\eql@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
226             \expandafter\@secondoftwo\eql@tmpc}}%
227         \fi
228       }%
229     \fi
230   }%
231   \ifdefined\eql@tmpd
232     \eql@tmpd

```

```

233 \else
234   \eq1@error{undefined value '#3' for option '#2' of '#1'}%
235 \fi
236 }

```

Decide between true and false or related pairs of values:

```

237 \def\eq1@decide@true{on,true,yes,enabled}
238 \def\eq1@decide@false{off,false,no,disabled}

```

`\eq1@decide@if`

```

239 \def\eq1@decide@if#1#2#3#4#5{%
240   \eq1@decide@select{#1}{#2}{#3}{%
241     {\eq1@decide@true{#4}},%
242     {\eq1@decide@false{#5}}}%

```

`\eq1@decide@bool` Store a boolean value into a conditional register:

```

243 \def\eq1@decide@bool#1#2#3#4{%
244   \eq1@decide@if{#1}{#2}{#3}{\let#4\eq1@true}{\let#4\eq1@false}}

```

### Key Declaration.

`\eq1@define@key` For convenience, we define a wrapper for keyval's `\define@key` which accepts lists of categories and keys. We prepend the prefix `eq1@` to all our categories so that we can hide it from the user in error messages:

```

245 \def\eq1@define@key#1#2{%
246   \eq1@ifnextchar@loose[%
247     {\eq1@definekey@opt{#1}{#2}}%
248     {\eq1@definekey@noopt{#1}{#2}}%
249 }
250 \def\eq1@definekey@noopt#1#2#3{\eq1@definekey@for{#1}{#2}{#3}}
251 \def\eq1@definekey@opt#1#2[#3]#4{\eq1@definekey@for{#1}{#2}{[#3]{#4}}}
252 \def\eq1@definekey@for#1#2#3{%
253   \def\eq1@for@fn##1##2##3{\define@key{eq1@##3}{##2}{#3}}%
254   \edef\eq1@for@vara{\noexpand\eq1@for@vara:=#1}%
255   \expandafter\@for\eq1@for@vara\do{%
256     \edef\eq1@for@varb{\noexpand\eq1@for@varb:=#2}%
257     \expandafter\@for\eq1@for@varb\do{%
258       \edef\eq1@for@call##1{%
259         \noexpand\eq1@for@fn{##1}{\eq1@for@varb}{\eq1@for@vara}}%
260       \eq1@for@call{##1}%
261     }%
262   }%
263 }

```

`\eq1@setkeys` Our wrapper of keyval's `\setkeys` prepends the prefix `eq1@` to the category, and it expands the list argument once:

```

264 \def\eq1@setkeys#1#2{%
265   \def\eq1@tmp{\setkeys{eq1@#1}}%
266   \expandafter\eq1@tmp\expandafter{#2}%
267 }

```

### Options and Control Interface.

`\eql@nextopt` It can be convenient to add arguments to the following equations environment, e.g.  
`\eql@nextopt@process` towards defining modifier macros:

```
268 \let\eql@nextopt\@empty
269 \def\eql@nextopt@process#1{%
270 (dev)\eql@dev@start\eql@nextopt@process
271   \eql@setkeys{#1}\eql@nextopt
272   \let\eql@tagging@opt\eql@nextopt
273   \global\let\eql@nextopt\@empty
274 }
```

`\eqnaddopt`

```
275 \newcommand{\eqnaddopt}[1]{%
276   \ifx\eql@nextopt\@empty
277     \eql@append\eql@nextopt{#1}%
278   \else
279     \eql@append\eql@nextopt{, #1}%
280   \fi
281 }
```

`\eqnlineset` Process global configuration options including the package options:

```
282 \newcommand{\eqnlineset}[1]{%
283 (dev)\eql@dev@start\eqnlineset
284   \eql@setkeys{setup}{#1}%
285   \ignorespaces
286 }
```

`\eql@control@default`

```
287 \protected\def\eql@control@default{%
288   \eql@warn@here\eqncontrol
289   \@gobble
290 }
291 \let\eqncontrol\eql@control@default
```

`\eqncontrol` Macro for general-purpose control within equations using key-value pairs:

```
292 \newcommand{\eql@control}[1]{%
293   \relax
294   \eql@setkeys{control}{#1}%
295   \ignorespaces
296 }
```

## C Parameters and Registers

In the following, we collect parameter and register definitions.

### C.1 Parameters

**TODO:** describe

**TODO:** maybe sort parameters into sections **TODO:** or sort parameters in sections here

`\eql@tagsleft` (*bool*) The boolean parameter `\eql@tagsleft` specifies whether the tags are placed at the left or right margin:

```
297 \let\eql@tagsleft\eql@false
```

`\eql@layoutleft` (*bool*) The boolean parameter `\eql@layoutleft` specifies whether to use left or central alignment layout:

```
298 \let\eql@layoutleft\eql@false
```

`\eql@layoutleftmargin` The default width of the left margin in left alignment layout is specified by `\eql@layoutleftmargin`. It may be pushed down to `\eql@layoutleftmarginmin` and up to `\eql@layoutleftmarginmax`:

```
299 \def\eql@layoutleftmargin{\leftmargini}
300 \def\eql@layoutleftmarginmax{.5\maxdimen}
301 \def\eql@layoutleftmarginmin{\z@}
```

`\eql@tagmargin@` (*dimen*) The intended margin width for tags in central alignment layout is stored in `\eql@tagmargin@` which is sourced by `\eql@tagmargin@val`. An undefined `\eql@tagmargin@val` will compute the margin width as the maximum width of tags (without separation). `\eql@tagmargin@ratio@` describes the maximum ratio of lines with tags to total number of lines for which `\eql@tagmargin@` is set to zero: **TODO:** threshold

```
302 \newdimen\eql@tagmargin@
303 \let\eql@tagmargin@val\@undefined
304 \newdimen\eql@tagmargin@ratio@
305 \eql@tagmargin@ratio@\p@
306 \def\eql@tagmargin@threshold{0.5}
```

`\eql@indent@` (*dimen*) The currently selected indentation width is specified by `\eql@indent@`. This dimension register is set to the macro `\eql@indent@val` when entering the equation environments:

```
307 \newdimen\eql@indent@
308 \def\eql@indent@val{2em}
```

`\eql@paddingleft@` (*dimen*) The padding of an equation (column) is specified by `\eql@paddingleft@` and `\eql@paddingright@`. These dimension registers are set to the macros `\eql@paddingleft@val` and `\eql@paddingright@val`, respectively, when entering the equation environments:

```
309 \newdimen\eql@paddingleft@
310 \newdimen\eql@paddingright@
311 \let\eql@paddingleft@val\@undefined
312 \let\eql@paddingright@val\@undefined
```

`\eql@display@linewidth` **TODO:** describe

```
313 \let\eql@display@linewidth\@undefined
314 \let\eql@display@marginleft\@undefined
315 \let\eql@display@marginright\@undefined
```

`\eql@box@colsep` The macro `\eql@box@colsep` specifies the intercolumn separation for equation boxes:

```
316 \def\eql@box@colsep{2em}
```

`\eql@spread@val` The extra spread of equation lines is specified by `\eql@spread@val`:

```

317 \def\eql@spread@val{\jot}
318 \newdimen\eql@spread@

```

`\eql@tagfuzz@` (*dimen*) The value `\eql@tagfuzz@` specifies the margin of error for comparing whether a tag fits a given equation line. We should not expect rounding errors in the fixed point arithmetic of T<sub>E</sub>X, nevertheless: **TODO**: probably do not need this due to fixed point arithmetic.

```

319 \newdimen\eql@tagfuzz@
320 \eql@tagfuzz@16sp\relax

```

`\eql@display@height` An equation will appear to the surrounding text with a fixed apparent height and depth specified by `\eql@display@height` and `\eql@display@depth`, respectively:

```

321 \def\eql@display@height\@undefined
322 \def\eql@display@depth\@undefined

```

`\eql@skip@mode@short` The setting `\eql@skip@mode@short` specifies when a reduced amount of glue should be used around equations in case the text line above the equation fits in the space that is left available in the first equation line. Value 0 turns this feature off, value 1 reduces the glue above the equation, value 2 furthermore reduces the glue below a single equation line and value 3 also reduces the glue below multi-line equations:

```

323 \def\eql@skip@mode@short{2}

324 \def\eql@skip@mode@cont@above{2}
325 \def\eql@skip@mode@cont@below{0}

326 \def\eql@skip@mode@par@above{3}
327 \def\eql@skip@mode@par@below{0}

328 \def\eql@skip@mode@top@above{4}
329 \def\eql@skip@mode@top@below{0}

330 \newcount\eql@skip@mode@leave@
331 \let\eql@skip@force@leave\@undefined

```

`\eql@skip@force@above` 0: short, 1: long, 2: cont, 3: par, 4: top, 5: no, 6: med, 7: custom

`\eql@skip@force@below`

`\eql@skip@mode@above@` (*counter*)

`\eql@skip@mode@below@` (*counter*)

```

332 \newcount\eql@skip@mode@above@
333 \newcount\eql@skip@mode@below@
334 \let\eql@skip@force@above\@undefined
335 \let\eql@skip@force@below\@undefined
336 \let\eql@skip@custom@above\@undefined
337 \let\eql@skip@custom@below\@undefined

```

`\eql@skip@cont@above` The glue when an equation is at the top of a horizontal list is specified by `\eql@skip@cont@above`:

`\eql@skip@top@above` The glue when an equation is at the top of a vertical list is specified by

`\eql@skip@top@below` `\eql@skip@top@above` and `\eql@skip@top@below`:

`\eql@skip@par@above` The glue when an equation starts a paragraph is specified by `\eql@skip@par@above`:

`\eql@skip@med@above` The surrounding glue for an equation with reduced spacing is given by

`\eql@skip@med@below` `\eql@skip@med@above` and `\eql@skip@med@below`:

```

338 \def\eq@skip@long@above{\abovedisplayskip}
339 \def\eq@skip@long@below{\belowdisplayskip}
340 \def\eq@skip@short@above{\abovedisplayshortskip}
341 \def\eq@skip@short@below{\belowdisplayshortskip}
342 \def\eq@skip@cont@above{\eq@skip@short@above}
343 \def\eq@skip@cont@below{\eq@skip@short@below}
344 \def\eq@skip@par@above{\eq@skip@long@above}
345 \def\eq@skip@par@below{\eq@skip@long@below}
346 \def\eq@skip@top@above{\eq@skip@long@above}
347 \def\eq@skip@top@below{\eq@skip@long@below}
348 \def\eq@skip@med@above{\abovedisplayskip/2}
349 \def\eq@skip@med@below{\belowdisplayskip/2}
350 \def\eq@skip@tag@above{\z@skip}
351 \def\eq@skip@tag@below{\z@skip}

```

`\eq@colsepmin@` (*dimen*) The minimum intercolumn separation is specified by `\eq@colsepmin@`. This dimension register is set to `\eq@colsepmin@val` when entering the equation environments to allow font-dependent values. Furthermore, `\eq@colsepmax@val` specifies the maximum intercolumn separation:

```

352 \newdimen\eq@colsepmin@
353 \def\eq@colsepmin@val{1em}
354 \def\eq@colsepmax@val{.5\maxdimen}

```

`\eq@tagwidthmin@` (*dimen*) The minimum tag width is specified by `\eq@tagwidthmin@`:

```

355 \newdimen\eq@tagwidthmin@
356 \eq@tagwidthmin@\z@

```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`.  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 's built-in value is half a quad<sup>5</sup> in font number 2. As the tag is processed in text mode, we use 0.5em instead.

```

357 \newdimen\eq@tagsepmin@
358 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

`\eq@equations@sqr@opt` Store the default arguments for `\[...\]` and `\<...\>`, respectively:

```

\eq@equations@ang@opt
\eq@box@ang@opt
359 \def\eq@equations@sqr@opt{equation,nonumber}
360 \def\eq@equations@ang@opt{align,nonumber}
361 \def\eq@box@ang@opt{align}

```

## Multi-Line Align Mode.

```

362 \let\eq@columns@fulllength\eq@false

```

## C.2 Registers

**TODO:** describe

**General.** **TODO:** describe

```

363 \newcount\eq@count@
364 \newdimen\eq@dimen@
365 \newskip\eq@skip@

```

---

<sup>5</sup>another half of a quad is left empty at the other end of the line.



**TODO:** describe

```
366 \let\eqldisplay@container\@empty
```

$\backslash\mathrm{eqldisplay@cellbox@}$  (*box*) The box  $\backslash\mathrm{eqldisplay@cellbox@}$  holds the present alignment component and  $\backslash\mathrm{eqldisplay@tagbox@}$  the tag for the present line. The corresponding dimensions  $\backslash\mathrm{eqldisplay@cellwidth@}$  and  $\backslash\mathrm{eqldisplay@tagwidth@}$  hold their widths.  $\backslash\mathrm{eqldisplay@prevwidth@}$  holds the width of the previous alignment component: **TODO:** adjust

```

\eqldisplay@cellwidth@ (dimen)
\eqldisplay@tagwidth@ (dimen)
\eqldisplay@prevdepth@ (dimen)
\eqldisplay@prevgraf@ (counter)
367 \newbox\eqldisplay@cellbox@
368 \newbox\eqldisplay@tagbox@
369 \newdimen\eqldisplay@cellwidth@
370 \newdimen\eqldisplay@prevwidth@
371 \newdimen\eqldisplay@tagwidth@
372 \newdimen\eqldisplay@prevdepth@
373 \newcount\eqldisplay@prevgraf@

```

```

\eqldisplay@totalwidth@ (dimen)
\eqldisplay@tagwidth@max@ (dimen)
\eqldisplay@totalheight@ (dimen)
374 \newdimen\eqldisplay@totalwidth@
375 \newdimen\eqldisplay@tagwidth@max@
376 \newdimen\eqldisplay@totalheight@
377 \newdimen\eqldisplay@topheight@
378 \newdimen\eqldisplay@bottomdepth@

```

$\backslash\mathrm{eqldisplay@line@height@}$  (*dimen*) The dimension registers  $\backslash\mathrm{eqldisplay@line@height@}$  and  $\backslash\mathrm{eqldisplay@line@depth@}$  keep track of the height and depth of the present line in an alignment:

```

379 \newdimen\eqldisplay@line@height@
380 \newdimen\eqldisplay@line@depth@

```

```

\eqldisplay@line@width@ (dimen)
\eqldisplay@line@avail@ (dimen)
\eqldisplay@line@pos@ (dimen)
\eqldisplay@line@widthsep@ (counter)
\eqldisplay@line@availsep@ (counter)
\eqldisplay@line@possep@ (counter)
\eqldisplay@line@offset@ (dimen)
\eqldisplay@line@prevdepth@ (dimen)
\eqldisplay@line@interline@ (dimen)
381 \newdimen\eqldisplay@line@width@
382 \newdimen\eqldisplay@line@avail@
383 \newdimen\eqldisplay@line@pos@
384 \newcount\eqldisplay@line@availsep@
385 \newcount\eqldisplay@line@widthsep@
386 \newcount\eqldisplay@line@possep@
387 \newdimen\eqldisplay@line@offset@
388 \newdimen\eqldisplay@line@prevdepth@
389 \newdimen\eqldisplay@line@interline@

```

## Rows and Columns.

$\backslash\mathrm{eqldisplay@row@}$  (*counter*) **TODO:** tagrows  $\backslash\mathrm{eqldisplay@row@}$  counts the present row (1-based) and  $\backslash\mathrm{eqldisplay@totalrows@}$  holds the total number of rows:

```

\eqldisplay@tagrows@ (counter)
390 \newcount\eqldisplay@row@
391 \newcount\eqldisplay@totalrows@
392 \newcount\eqldisplay@tagrows@

```

```

\eqldisplay@column@
\eqldisplay@totalcolumns@
393 \newcount\eqldisplay@column@
394 \newcount\eqldisplay@totalcolumns@

```

`\eq@colsep@` (*dimen*) The dimension of the intercolumn separation for align environments is stored in `\eq@colsep@`:

```
395 \newdimen\eq@colsep@
```

`intercolumns@` (*counter*)

```
396 \newcount\eq@intercolumns@
```

## Vertical Spacing Adjustments.

`\firstavail@` (*dimen*) The unused space on the first line of an alignment is stored in `\eq@display@firstavail@` for comparison against `\predisplaysize` and determining short skip mode of display equations. It is convenient to set it via `\eq@display@firstavail@set` provided that we are on the first line:

```
397 \newdimen\eq@display@firstavail@
398 \def\eq@display@firstavail@set#1{%
399   \ifnum\eq@row@=\@one
400     \global\eq@appendexpand\eq@display@container{%
401       \eq@display@firstavail@the#1\relax}%
402   \fi
403 }
```

The counter stores whether the tag one first/last line is raised/lowered as 1/2 (or 3 for both). This implies a different vskip corresponding to the mostly empty line: **TODO:** adjust

```
404 \newdimen\eq@display@aboveextend@
405 \newdimen\eq@display@belowextend@
```

## Shared Registers.

`\ifmeasuring@` (*bool*) All display environments get typeset twice – once during a “measuring” phase and then again during a “production” phase. We reuse the original `amsmath` definition `\ifmeasuring@` to determine which case we’re in, so we and other packages may take appropriate action. It does not hurt to define this conditional in any case. We should tell `hyperref` about measuring processes as we’re not `amsmath` and not being catered for:

```
406 \ifdefined\measuring@true\else
407   \expandafter\newif\csname ifmeasuring@\endcsname
408 \fi
409 \AddToHook{package/hyperref/after}{
410   \ifdefined\Hy@ifnotmeasuring
411     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
412   \fi
413 }
```

`\if@display` (*bool*) `amsmath` defines the conditional `\if@display` to test whether we’re in a display equation including the inner math parts of equation blocks. We provide it in case `amsmath` is absent, and initialise it:

```
414 \ifdefined\@displaytrue\else
415   \expandafter\newif\csname if@display\endcsname
416   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
417 \fi
```

## C.3 Hooks

`\eql@hook@...` For what it's worth, we define a couple of entry points where one might hook into the equations typesetting framework. The L<sup>A</sup>T<sub>E</sub>X hook framework would be more versatile, but as the purpose of these hooks is rather unclear at the moment, we make this as efficient as it could get: **TODO:** may add a few more hooks

```
418 \let\eql@hook@blockbefore\@empty
419 \let\eql@hook@blockafter\@empty
420 \let\eql@hook@blockin\@empty
421 \let\eql@hook@blockout\@empty
422 \let\eql@hook@linein\@empty
423 \let\eql@hook@lineout\@empty
424 \let\eql@hook@colin\@empty
425 \let\eql@hook@colout\@empty
426 \let\eql@hook@eqin\@empty
427 \let\eql@hook@eqout\@empty
428 \let\eql@hook@innerleft\@empty
429 \let\eql@hook@innerright\@empty
430 \let\eql@hook@innerlead\@empty
```

## D Features

### D.1 Punctuation

The equations environments supply an automatic punctuation scheme which allows to define a default punctuation at the end of each column, line and equation block.

`\eql@punct@col` These macros store the punctuation character for columns, lines and blocks. A value  
`\eql@punct@line` `\relax` indicates that the punctuation should be handed down to the next lower level:  
`\eql@punct@block` **TODO:** update

```
431 \let\eql@punct@col\@empty
432 \let\eql@punct@line\relax
433 \let\eql@punct@block\relax
434 \let\eql@punct@main\relax
```

`\eql@punct@sep` This macro stores the separation to be applied before the punctuation (unless it is empty):

```
435 \let\eql@punct@sep\relax
```

`\eql@punct@set` **TODO:** describe

```
436 \def\eql@punct@relax{\relax}
437 \def\eql@punct@tilde{~}
438 \def\eql@punct@set#1#2{%
439   \def#1{#2}%
440   \ifx#1\eql@punct@relax
441     \let#1\relax
442   \fi
443   \ifx#1\eql@punct@tilde
444     \let#1\@empty
445   \fi
446 }
```

`\eqnpunct` Set the punctuation for columns, lines and blocks. Note that the macro `\eqnpunct` sets the punctuation for the following equation block or for the current cell. Starred versions clear the punctuation for the respectively levels:

```

447 \def\eqnpunct{%
448   \eq@ifstar@tight\eq@punct@next@setrelax\eq@punct@next@set}
449 \def\eq@punct@next@set#1{%
450   \ifmmode
451     \eq@punct@set\eq@punct@col{#1}%
452     \eq@punct@set\eq@punct@line{#1}%
453     \eq@punct@set\eq@punct@block{#1}%
454   \else
455     \eqnadopt{punct={#1}}%
456   \fi
457   \ignorespaces}
458 \def\eq@punct@next@setrelax{%
459   \ifmmode
460     \let\eq@punct@block\relax
461   \else
462     \eqnadopt{punct*}%
463   \fi
464   \ignorespaces}

```

`\eq@punct@apply@col` Output the punctuation for the present column. If non-empty, prepend some separation. Clear the punctuation so that no further column punctuation is output within the current group:

```

465 \def\eq@punct@apply@col{%
466   \ifx\eq@punct@col\@empty\else
467     \eq@punct@sep
468     \eq@punct@col
469     \let\eq@punct@col\@empty
470   \fi
471 }

```

Output the punctuation currently set for lines unless disabled. Alike `\eq@punct@apply@col` prevent further output of punctuations for lines and columns within the current group:

`\eq@punct@apply@line`

```

472 \def\eq@punct@apply@line{%
473   \ifx\eq@punct@line\relax
474   % \TODO hand down immediately?
475   \else
476     \ifx\eq@punct@line\@empty\else
477       \eq@punct@sep
478       \eq@punct@line
479     \fi
480     \let\eq@punct@line\relax
481     \let\eq@punct@col\@empty
482   \fi
483 }

```

`\eq@punct@apply@block` Outputs the punctuation for the current equation block unless disabled in analogy to `\eqnpunctapply` `\eq@punct@apply@line`:

```

484 \def\eq@punct@apply@block{%
485   \ifx\eq@punct@block\relax

```

```

486 % \TODO hand down immediately?
487 \else
488   \ifx\eql@punct@block\@empty\else
489     \eql@punct@sep
490     \eql@punct@block
491   \fi
492   \let\eql@punct@block\relax
493   \let\eql@punct@line\relax
494   \let\eql@punct@col\@empty
495 \fi
496 }

497 \let\eqnpunctapply\eql@punct@apply@block

```

## D.2 Math Classes at Alignment

The following describes the adjustment of math classes surrounding the alignment marker.

`\class@innerright@sel@` Select between `\eql@class@innerlead` and `\eql@class@innerright` depending on whether the left part of the aligned column is empty:

```

498 \def\eql@class@innerright@sel@{%
499   \ifdim\eql@prevwidth@=\z@
500     \eql@class@innerlead
501   \else
502     \eql@class@innerright
503   \fi
504 }

```

`\class@innerleft@set` Set the left, right and leading math classes. Setting the right math class disables the leading math class, so the leading math class must be specified after the right one:

```

\class@innerright@set
\class@innerlead@set
505 \def\eql@class@innerleft@set#1{%
506   \def\eql@class@innerleft{#1}%
507 }
508 \def\eql@class@innerright@set#1{%
509   \def\eql@class@innerright{#1}%
510   \let\eql@class@innerright@sel\eql@class@innerright
511 }
512 \def\eql@class@innerlead@set#1{%
513   \def\eql@class@innerlead{#1}%
514   \let\eql@class@innerright@sel\eql@class@innerright@sel@
515 }

```

`\eql@class@ampeq` We define two standard combinations of math classes intended to be used with ‘&=’ (ampeq) or ‘=&’ (eqamp). The default setting is ‘&=’ (ampeq):

```

516 \def\eql@class@ampeq{%
517   \eql@class@innerleft@set{}%
518   \eql@class@innerright@set{}}%
519 }
520 \def\eql@class@eqamp{%
521   \eql@class@innerleft@set{\mathrel{}}%
522   \eql@class@innerright@set{\mathrel{}}%
523   \eql@class@innerlead@set{}}%
524 }
525 \eql@class@ampeq

```

## D.3 Framed Cells

**TODO:** describe **TODO:** warn if issued in even cells

```
526 \let\eql@frame@cmd\@undefined
527 \newdimen\eql@frame@margin@
528 \def\eql@frame@set[#1]{%
529   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
530   \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbbox}
531   \def\eql@frame@measure{%
532     \setbox\z@\hbox{\eql@frame@cmd{}}%
533     \eql@frame@margin@.5\wd\z@
534   }
535   \def\eql@frame@print{%
536     \setbox\eql@cellbox@\hbox{%
537       \eql@frame@cmd{\unhbox\eql@cellbox@}%
538     }%
539   }
540   \def\eql@frame@adjust{%
541     \setbox\eql@cellbox@\hbox{%
542       \eql@frame@cmd{%
543         \unhbox\eql@cellbox@
544         \unkern
545         \unskip
546       }%
547       \hfil
548       \kern\z@
549     }%
550   }
```

## D.4 Alternative Content Description

**TODO:** describe **TODO:** would be nice to provide as environments as well **TODO:** implement for PDF tagging

```
551 \DeclareRobustCommand{\eqnalt}[2][]{}
```

# E Equation Numbering

**TODO:** describe

## E.1 Supporting Definitions

Parameters.

```
552 \let\eql@tags@autolabel\eql@false
553 \let\eql@tags@autotag\eql@true
554 \let\eql@tags@warn\eql@true

555 \def\eql@tags@name@generic{[equation]}

556 \let\eql@tagpos@doconvert\eql@false

557 \def\eql@tagpos@snap{4pt}
```

## Registers.

```
558 \let\eql@numbering@mode\@undefined

559 \let\eql@numbering@active\eql@true
560 \let\eql@numbering@multi\eql@true

561 \let\eql@tags@container\@undefined
562 \def\eql@tags@container@clear{%
563   \let\eql@tags@label\@undefined
564   \let\eql@tags@name\@undefined
565   \let\eql@tags@tag\@undefined
566   \let\eql@tags@ref\@undefined
567   \let\eql@tags@anchor\@empty
568   \eql@tagpos@shift@z@
569   \eql@tagpos@smashup@z@
570   \eql@tagpos@smashdown@z@
571   \let\eql@tagpos@reserve\eql@true
572 }

573 \let\eql@tags@label\@undefined
574 \let\eql@tags@name\@undefined
575 \let\eql@tags@tag\@undefined
576 \let\eql@tags@ref\@undefined
577 \let\eql@tags@frame@cmd\@firstofone
```

tags@glabel@ (*counter*)

```
578 \newcount\eql@tags@glabel@
579 \eql@tags@glabel@z@
580 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
581 \def\eql@tags@glabel@step{\global\advance\eql@tags@glabel@\@ne}

582 \let\eql@tagpos@continuous\eql@false

583 \newcount\eql@tagpos@row@
584 \newcount\eql@tagpos@prevrow@
585 \newdimen\eql@tagpos@shift@
586 \newdimen\eql@tagpos@smashup@
587 \newdimen\eql@tagpos@smashdown@
588 \newdimen\eql@tagpos@current@
589 \newdimen\eql@tagpos@plain@
590 \newdimen\eql@tagpos@raised@
591 \newdimen\eql@tagpos@target@
592 \newdimen\eql@tagpos@headroom@
593 \newdimen\eql@tagpos@footroom@
```

## E.2 Schemes

**TODO:** describe

### Table.

```
594 \def\eql@numbering@tab@sub{sub}
595 \def\eql@numbering@tab@all{all}
596 \def\eql@numbering@tab@first{first}
597 \def\eql@numbering@tab@last{last}
598 \def\eql@numbering@tab@in{in}
```

```

599 \def\eqL@numbering@tab@out{out}
600 \def\eqL@numbering@tab@middle{middle}
601 \def\eqL@numbering@tab@best{best}
602 \def\eqL@numbering@tab@here{here}
603 \def\eqL@numbering@tab@top{top}
604 \def\eqL@numbering@tab@bottom{bottom}
605 \def\eqL@numbering@tab@center{center}
606 \def\eqL@numbering@tab@centerone{centerone}
607 \def\eqL@numbering@tab@median{median}
608 \def\eqL@numbering@tab@baseline{baseline}

609 \let\eqL@numbering@mode\eqL@numbering@tab@all
610 \let\eqL@numbering@mode@multi\eqL@numbering@tab@all
611 \let\eqL@numbering@mode@single\eqL@numbering@tab@out

```

**TODO:** describe

```

612 \let\eqL@numbering@tab@subeq\eqL@numbering@tab@sub
613 \let\eqL@numbering@tab@subequation\eqL@numbering@tab@sub
614 \let\eqL@numbering@tab@subequations\eqL@numbering@tab@sub
615 \let\eqL@numbering@tab@mid\eqL@numbering@tab@middle
616 \let\eqL@numbering@tab@outside\eqL@numbering@tab@out
617 \let\eqL@numbering@tab@inside\eqL@numbering@tab@in
618 \let\eqL@numbering@tab@within\eqL@numbering@tab@in
619 \let\eqL@numbering@tab@opt\eqL@numbering@tab@best
620 \let\eqL@numbering@tab@optimal\eqL@numbering@tab@best
621 \let\eqL@numbering@tab@pick\eqL@numbering@tab@here
622 \let\eqL@numbering@tab@med\eqL@numbering@tab@median
623 \eqL@letcs{eqL@numbering@tab@center*}\eqL@numbering@tab@baseline
624 \eqL@letcs{eqL@numbering@tab@center!}\eqL@numbering@tab@centerone

```

**TODO:** describe

```

625 \let\eqL@numbering@tab@a\eqL@numbering@tab@all
626 \let\eqL@numbering@tab@s\eqL@numbering@tab@sub
627 \let\eqL@numbering@tab@f\eqL@numbering@tab@first
628 \let\eqL@numbering@tab@l\eqL@numbering@tab@last
629 \let\eqL@numbering@tab@o\eqL@numbering@tab@out
630 \let\eqL@numbering@tab@i\eqL@numbering@tab@in
631 \let\eqL@numbering@tab@h\eqL@numbering@tab@here
632 \let\eqL@numbering@tab@t\eqL@numbering@tab@top
633 \let\eqL@numbering@tab@b\eqL@numbering@tab@bottom
634 \let\eqL@numbering@tab@c\eqL@numbering@tab@center
635 \let\eqL@numbering@tab@m\eqL@numbering@tab@median
636 \eqL@letcs{eqL@numbering@tab@+}\eqL@numbering@tab@best
637 \eqL@letcs{eqL@numbering@tab@m*}\eqL@numbering@tab@middle
638 \eqL@letcs{eqL@numbering@tab@c*}\eqL@numbering@tab@baseline
639 \eqL@letcs{eqL@numbering@tab@c!}\eqL@numbering@tab@centerone

```

**Implementations.** **TODO:** describe

```

640 \def\eqL@numbering@init@all{\let\eqL@numbering@mode\eqL@true}

```

**TODO:** describe

```

641 \def\eqL@numbering@init@sub{%
642   \let\eqL@numbering@mode\eqL@true
643   \ifdefined\eqL@subequations@active
644     \let\eqL@numbering@mode\eqL@numbering@tab@all
645   \else

```



```

646 \let\eq@numbering@subeq@use\eq@true
647 \fi
648 }

649 \def\eq@numbering@init@first{\eq@tagpos@row@{\@ne}
650 \def\eq@numbering@init@last{\eq@tagpos@row@{\@MM}
651 \def\eq@numbering@init@here{\eq@tagpos@row@{\m@ne}

```

**TODO:** describe

```

652 \def\eq@numbering@init@in{%
653 \ifdefined\eq@tagsleft
654 \eq@numbering@init@last
655 \else
656 \eq@numbering@init@first
657 \fi
658 }

```

**TODO:** describe

```

659 \def\eq@numbering@init@out{%
660 \ifdefined\eq@tagsleft
661 \eq@numbering@init@first
662 \else
663 \eq@numbering@init@last
664 \fi
665 }

```

**TODO:** describe

```

666 \def\eq@tagpos@eval@middle{%
667 \ifnum\eq@tagpos@row@=\z@
668 \eq@tagpos@row@{\numexpr(\eq@totalrows@
669 +\ifdefined\eq@tagsleft\z@{\else\@ne}\fi)/\tw@\relax
670 \fi
671 }

```

**TODO:** describe

```

672 \def\eq@tagpos@eval@best{%
673 \ifnum\eq@tagpos@row@=\z@
674 \let\eq@numbering@best@use\eq@true
675 \eq@numbering@init@out
676 \fi
677 }

```

**TODO:** describe

```

678 \def\eq@numbering@init@continuous{\let\eq@tagpos@continuous\eq@true}

```

**TODO:** describe

```

679 \let\eq@numbering@init@top\eq@numbering@init@continuous
680 \def\eq@tagpos@eval@top{%
681 \eq@tagpos@current@\z@
682 }

```

**TODO:** describe

```

683 \let\eq@numbering@init@bottom\eq@numbering@init@continuous
684 \def\eq@tagpos@eval@bottom{%
685 \eq@tagpos@current@\dimexpr\eq@totalheight@
686 -\eq@tagheight@block@-\eq@tagdepth@block@\relax
687 }

```

**TODO:** describe

```
688 \let\eql@numbering@init@center\eql@numbering@init@continuous
689 \def\eql@tagpos@eval@center{%
690   \ifnum\eql@totalrows@=\@ne
691     \eql@tagpos@row@\@ne
692   \fi
693   \eql@tagpos@current@\dimexpr(\eql@totalheight@
694     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
695 }
```

**TODO:** describe

```
696 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
697 \def\eql@tagpos@eval@centerone{%
698   \eql@tagpos@current@\dimexpr(\eql@totalheight@
699     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
700 }
```

**TODO:** describe

```
701 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
702 \def\eql@tagpos@eval@baseline{%
703   \eql@tagpos@current@\dimexpr(\eql@totalheight@
704     +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
705 }
```

**TODO:** describe

```
706 \let\eql@numbering@init@median\eql@numbering@init@continuous
707 \def\eql@tagpos@eval@median{%
708   \ifnum\eql@tagpos@row@=\z@
709     \ifodd\eql@totalrows@
710       \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
711     \else
712       \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
713       \eql@dimensions@get\eql@tagpos@row@
714       \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
715         +(\eql@line@interline@-\eql@tagheight@block@
716         +\eql@tagdepth@block@)/\tw@\relax
717     \fi
718     \ifnum\eql@totalrows@=\@ne
719       \eql@tagpos@row@\@ne
720     \else
721       \eql@tagpos@adjust@eval@convert
722       \eql@tagpos@row@\z@
723     \fi
724   \fi
725 }
```

**Selection.**

```
726 \def\eql@numbering@set#1{%
727   \ifcsname eql@numbering@tab@#1\endcsname
728     \expandafter\let\expandafter\eql@numbering@mode
729       \csname eql@numbering@tab@#1\endcsname
730   \ifx\eql@numbering@mode\eql@numbering@tab@all
731     \let\eql@numbering@mode@multi\eql@numbering@mode
732   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
733     \let\eql@numbering@mode@multi\eql@numbering@mode
```

```

734 \else
735 \let\eql@numbering@mode@single\eql@numbering@mode
736 \fi\fi
737 \else
738 \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
739 \let\eql@numbering@mode\eql@numbering@tab@all
740 \fi
741 }

```

**TODO:** describe

```

742 \def\eql@numbering@init{%
743 \let\eql@numbering@multi\eql@false
744 \let\eql@tagpos@continuous\eql@false
745 \let\eql@numbering@subeq@use\eql@false
746 \let\eql@numbering@best@use\eql@false
747 \eql@tagpos@row@z@
748 \csname eql@numbering@init@\eql@numbering@mode\endcsname
749 \ifdefined\eql@numbering@active
750 \let\eql@numbering@eqnswinit\@eqnswtrue
751 \else
752 \let\eql@numbering@eqnswinit\@eqnswfalse
753 \fi
754 \let\eql@numbering@active\eql@false
755 }

```

### E.3 Interface

**Activation.** **TODO:** note \nonumber already defined, modifications by amsmath

```

756 \eql@amsmath@after{
757 \let\eql@print@eqnum@default\print@eqnum
758 \let\eql@incr@eqnum@default\incr@eqnum
759 }

```

**TODO:** describe

```

760 \protected\def\donumber{%
761 \if@eqnsw\else
762 \global\@eqnswtrue
763 \ifx\print@eqn\@empty
764 \global\let\print@eqn\eql@print@eqnum@default
765 \fi
766 \ifx\incr@eqn\@empty
767 \global\let\incr@eqn\eql@incr@eqnum@default
768 \fi
769 \fi
770 }

```

**TODO:** reconsider operation

\numberhere

```

771 \protected\def\eql@numberhere{%
772 \ifdefined\eql@numbering@multi
773 \global\@eqnswtrue
774 \else
775 \global\eql@tagpos@row@\eql@row@
776 \fi
777 }

```

**TODO:** describe

`\numbernext`

```
778 \protected\def\eq1@numbernext{%
779   \ifdefined\eq1@numbering@multi
780     \global\@eqnswfalse
781   \else
782     \ifnum\eq1@tagpos@row@=\eq1@row@
783       \global\advance\eq1@tagpos@row@\@ne
784     \fi
785   \fi
786 }
```

**Activation Trigger.**

```
787 \def\eq1@tags@autoenable{%
788   \global\@eqnswtrue
789   \ifnum\eq1@tagpos@row@=\m@ne
790     \numberhere
791   \fi
792 }
```

**Labels.** **TODO:** describe

`\eq1@label@org`

```
793 \let\eq1@label@org\label
```

**TODO:** describe

```
794 \def\eq1@label@gobble{\eq1@ampprotect\eq1@testopt@tight\eq1@gobbleoptone{}}
```

**TODO:** describe

```
795 \protected\def\eq1@label{%
796   \eq1@ampprotect\eq1@testopt@tight\eq1@tags@add@labelname\eq1@testopt@default
797 }
```

**TODO:** describe

```
798 \def\eq1@tags@add@labelname[#1]#2{%
799   \def\eq1@tmp{#1}%
800   \ifx\eq1@tmp\eq1@testopt@default\else
801     \eq1@tags@add@name{#1}%
802   \fi
803   \eq1@tags@add@label{#2}%
804 }
```

**TODO:** describe

```
805 \def\eq1@tags@set@label#1{%
806   \ifdefined\eq1@tags@warn
807     \ifdefined\eq1@tags@label
808       \eq1@warn@label@multiple{#1}%
809     \fi
810   \fi
811   \def\eq1@tags@label{#1}%
812 }
```

**TODO:** describe

```
813 \def\eql@tags@set@name#1{%
814   \ifdefined\eql@tags@warn
815     \ifdefined\eql@tags@name
816       \eql@warn@name@multiple
817     \fi
818   \fi
819   \def\eql@tags@name{#1}%
820 }
```

**TODO:** describe

```
821 \def\eql@tags@add@label#1{%
822   \ifdefined\eql@tags@autolabel
823     \eql@tags@autoenable
824   \fi
825   \global\eql@appendexpand\eql@tags@container{%
826     \noexpand\eql@tags@set@label{#1}}%
827 }
```

**TODO:** describe

```
828 \def\eql@tags@add@name#1{%
829   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
830   \global\eql@appendmacro\eql@tags@container\eql@tmp
831 }
```

**TODO:** describe

```
832 \def\eql@tags@addblock@label#1{%
833   \eql@appendexpand\eql@tags@container@block{%
834     \noexpand\eql@tags@set@label{#1}}%
835 }
```

**TODO:** describe

```
836 \def\eql@tags@addblock@name#1{%
837   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
838   \eql@appendmacro\eql@tags@container@block\eql@tmp
839 }
```

**Tags.** **TODO:** describe

`\eql@tag@default`

```
840 \protected\def\eql@tag@default{%
841   \eql@warn@here\tag
842   \eql@tag@gobble
843 }
844 \let\tag\eql@tag@default
```

`\eql@tag@gobble`

```
845 \def\eql@tag@gobble{%
846   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}
```

**TODO:** describe

```
847 \protected\def\eql@tag{%
848   \eql@ampprotecttwo\eql@teststaropt@tight
```

```

849     {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
850     \eql@testopt@default
851 }

```

\eql@tags@add@tagref

```

852 \def\eql@tags@add@tagref[#1]#2{%
853   \def\eql@tmp{#1}%
854   \ifx\eql@tmp\eql@testopt@default\else
855     \eql@tags@add@ref{#1}%
856   \fi
857   \eql@tags@add@tag{#2}%
858 }

```

**TODO:** describe

```

859 \def\eql@tags@set@tag#1{%
860   \ifdefined\eql@tags@warn
861     \ifdefined\eql@tags@tag
862       \eql@warn@tag@multiple
863     \fi
864   \fi
865   \def\eql@tags@tag{#1}%
866 }

```

**TODO:** describe

```

867 \def\eql@tags@set@ref#1{%
868   \ifdefined\eql@tags@warn
869     \ifdefined\eql@tags@ref
870       \eql@warn@ref@multiple
871     \fi
872   \fi
873   \def\eql@tags@ref{#1}%
874 }

```

**TODO:** describe

```

875 \def\eql@tags@add@tag#1{%
876   \ifdefined\eql@tags@autotag
877     \eql@tags@autoenable
878   \fi
879   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
880   \global\eql@appendmacro\eql@tags@container\eql@tmp
881 }

```

**TODO:** describe

```

882 \def\eql@tags@add@ref#1{%
883   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
884   \global\eql@appendmacro\eql@tags@container\eql@tmp
885 }

```

tags@add@tagform@off

```

886 \def\eql@tags@add@tagform@off{%
887   \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
888 }

```

**TODO:** describe

```

889 \def\eql@tags@addblock@tag#1{%

```

```

890 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
891 \eql@appendmacro\eql@tags@container@block\eql@tmp
892 }

```

**TODO:** describe

```

893 \def\eql@tags@addblock@ref#1{%
894 \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
895 \eql@appendmacro\eql@tags@container@block\eql@tmp
896 }

```

**TODO:** describe

```

897 \def\eql@tags@addblock@tagform@off{%
898 \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
899 }

```

## Raise Tags.

\raisetag

```

900 \def\eql@raisetag@default{%
901 \eql@warn@here\raisetag
902 \eql@raisetag@gobble
903 }

904 \def\eql@raisetag@gobble{%
905 \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble
906 }

```

**TODO:** describe

```

907 \eql@amsmath@let\raisetag\eql@raisetag@default

908 \def\eql@raisetag{%
909 \eql@ampprotecttwo\eql@ifstar@tight\eql@tags@add@raiseshift\eql@raisetag@test
910 }

911 \def\eql@raisetag@test#1{%
912 \def\eql@tmpa{#1}\def\eql@tmpb{!}%
913 \ifx\eql@tmpa\eql@tmpb
914 \eql@tags@add@forceraise
915 \else
916 \eql@tags@add@raisesmash{#1}%
917 \fi
918 }

919 \def\eql@tags@add@raiseshift#1{%
920 \global\eql@appendexpand\eql@tags@container{%
921 \advance\eql@tagpos@shift@the\glueexpr#1\relax\relax}%
922 }

923 \def\eql@tags@add@raisesmash#1{%
924 \dimen@glueexpr#1\relax
925 \ifdim\dimen@<\z@
926 \global\eql@appendexpand\eql@tags@container{%
927 \advance\eql@tagpos@smashdown@the\dimen@\relax}%
928 \else
929 \global\eql@appendexpand\eql@tags@container{%
930 \advance\eql@tagpos@smashup@the\dimen@\relax}%
931 \fi
932 }

```

```

933 \def\eql@tags@add@forceraise{%
934   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
935 }

```

## E.4 Integration

**TODO:** describe

**Support.** **TODO:** describe

```

936 \def\eql@numbering@settools{%
937   \let\label\eql@label
938   \let\tag\eql@tag
939   \let\raisetag\eql@raisetag
940   \let\numberhere\eql@numberhere
941   \let\numbernext\eql@numbernext
942 }

```

**TODO:** not necessary anymore

```

943 \def\eql@numbering@settools@gobble{%
944   \let\label\eql@label@gobble
945   \let\tag\eql@tag@gobble
946   \let\raisetag\eql@raisetag@gobble
947   \let\numberhere\relax
948   \let\numbernext\relax
949 }

```

```

950 \def\eql@numbering@autoblock{%
951   \begingroup
952     \let\eql@tags@warn\eql@false
953     \eql@tags@container@block
954     \ifdefined\eql@tags@autolabel
955       \ifdefined\eql@tags@label
956         \global\@eqnswtrue
957       \fi
958     \fi
959     \ifdefined\eql@tags@autotag
960       \ifdefined\eql@tags@tag
961         \global\@eqnswtrue
962       \fi
963     \fi
964   \endgroup
965 }

```

```

966 \def\eql@numbering@warnunused{%
967   \ifdefined\eql@tags@label
968     \eql@warn@label@unused
969   \fi
970   \ifdefined\eql@tags@name
971     \eql@warn@name@unused
972   \fi
973   \ifdefined\eql@tags@tag
974     \eql@warn@tag@unused
975   \fi
976   \ifdefined\eql@tags@erf
977     \eql@warn@ref@unused
978   \fi

```



979 }

**Single Line. TODO:** describe

```
980 \def\eql@numbering@single@init{%
981   \let\eql@numbering@multi\eql@false
982   \eql@numbering@settools
983   \eql@numbering@eqnswinit
984   \eql@numbering@autoblock
985   \global\let\eql@tags@container\eql@tags@container@block
986   \let\eql@tags@warn\eql@true
987 }

988 \def\eql@numbering@single@eval{%
989   \ifnum\eql@tagpos@row@=\m@ne
990     \eqnswfalse
991   \fi
992 }
```

**Multi-Line Measuring Pass. TODO:** describe

```
993 \def\eql@numbering@measure@init{%
994   \eql@numbering@settools
995   \ifdefined\eql@numbering@multi\else
996     \eql@numbering@eqnswinit
997     \eql@numbering@autoblock
998   \fi
999   \global\let\eql@tags@container\eql@tags@container@block
1000   \let\eql@tags@warn\eql@true
1001 }
```

**TODO:** might select only relevant routines in init **TODO:** describe

```
1002 \def\eql@numbering@measure@line@begin{%
1003   \ifdefined\eql@numbering@multi
1004     \global\eql@numbering@eqnswinit
1005   \fi
1006 }
```

**TODO:** describe

```
1007 \def\eql@numbering@measure@blocktag{%
1008   \ifdefined\eql@numbering@multi
1009     \eqnswfalse
1010   \else
1011     \ifnum\eql@tagpos@row@=\m@ne
1012       \eqnswfalse
1013     \fi
1014     \ifnum\eql@totalrows@=\z@
1015       \eqnswfalse
1016     \fi
1017   \fi
1018 }
```

**Multi-Line Print Pass. TODO:** describe

**TODO:** can we be absolutely sure about all values being preserved global might pick up a value from a higher level block and restore it globally!

```

1019 \def\eql@numbering@print@init{%
1020   \let\eql@tags@warn\eql@false
1021   \ifdefined\eql@numbering@multi
1022     \eql@numbering@settools
1023     \global\let\eql@tags@container\eql@tags@container@block
1024   \else
1025     \let\eql@tags@container@block\eql@tags@container
1026     \eql@numbering@settools@gobble
1027   \fi
1028 }

```

**TODO:** might select only relevant routines in init **TODO:** describe

```

1029 \def\eql@numbering@print@block@begin{%
1030   \ifdefined\eql@numbering@multi\else
1031     \ifnum\eql@tagpos@row@>\z@
1032       \eql@tags@makeblockanchor
1033       \global\eql@appendexpand\eql@tags@container@block{%
1034         \def\noexpand\eql@tags@anchor{%
1035           \unexpanded\expandafter{\eql@tags@anchor}}}%
1036     \fi
1037   \fi
1038   \ifdefined\eql@numbering@subeq@use
1039     \eql@tags@printsubeqlabel
1040   \fi
1041 }

```

**TODO:** describe

```

1042 \def\eql@numbering@print@line@begin{%
1043   \ifdefined\eql@numbering@multi
1044     \global\eql@numbering@eqnswinit
1045   \fi
1046 }

```

**TODO:** describe

```

1047 \def\eql@numbering@print@line@eval{%
1048   \ifdefined\eql@numbering@multi
1049     \if@eqnsw
1050       \eql@tags@container
1051     \fi
1052   \else
1053     \ifnum\eql@tagpos@row@=\eql@row@
1054       \@eqnswtrue
1055       \eql@tags@container@block
1056     \else
1057       \@eqnswfalse
1058     \fi
1059   \fi
1060 }

```

## E.5 Positioning

**TODO:** describe

```

1061 \def\eql@tagpos@single@eval{%
1062   \if@eqnsw
1063     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1064     \ifnum\eql@tagpos@row@>\@ne

```

```

1065     \eql@tagpos@row@\@ne
1066     \fi
1067     \ifdefined\eql@tagpos@doconvert
1068         \let\eql@tagpos@continuous\eql@true
1069     \fi
1070     \ifdefined\eql@tagpos@continuous
1071         \eql@tagpos@single@eval@continuous
1072     \fi
1073 \else
1074     \eql@tagpos@row@\z@
1075 \fi
1076 \eql@tagpos@prevrow@\z@
1077 \eql@tagpos@headroom@\z@
1078 \eql@tagpos@footroom@\z@
1079 }

```

**TODO:** describe

```

1080 \def\eql@tagpos@single@eval@continuous{%
1081     \ifnum\eql@tagpos@row@>\z@
1082         \eql@tagpos@target@\eql@tagpos@shift@
1083     \else
1084         \eql@tagpos@target@\dimexpr\eql@line@height@
1085             -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1086     \fi
1087     \eql@tagpos@row@\@ne
1088     \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1089         \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1090     \eql@tagpos@target@\z@
1091 \fi
1092 }

```

**TODO:** describe

```

1093 \def\eql@tagpos@adjust@eval{%
1094     \if@eqnsw
1095         \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1096         \ifnum\eql@tagpos@row@>\eql@totalrows@
1097             \eql@tagpos@row@\eql@totalrows@
1098         \fi
1099         \ifdefined\eql@tagpos@doconvert
1100             \let\eql@tagpos@continuous\eql@true
1101         \fi
1102         \ifdefined\eql@tagpos@continuous
1103             \ifnum\eql@tagpos@row@>\z@
1104                 \eql@tagpos@adjust@eval@convert
1105             \fi
1106             \eql@tagpos@adjust@eval@continuous
1107         \fi
1108     \else
1109         \eql@tagpos@row@\z@
1110         \eql@tagpos@prevrow@\z@
1111     \fi
1112 }

```

**TODO:** describe

```

1113 \def\eql@tagpos@adjust@eval@convert{%
1114     \eql@tagpos@current@\z@
1115     \eql@dimensions@for{%
1116         \ifnum\eql@row@<\eql@tagpos@row@

```

```

1117 \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1118 +\eql@line@height@+\eql@line@depth@\relax
1119 \fi
1120 \ifnum\eql@row@=\eql@tagpos@row@
1121 \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1122 +\eql@line@height@-\eql@tagheight@block@\relax
1123 \fi
1124 }%
1125 }

```

**TODO:** describe

```

1126 \def\eql@tagpos@adjust@eval@continuous{%
1127 \dimen@\dimexpr\eql@tagpos@current@-\eql@tagpos@shift@\relax
1128 \eql@tagpos@row@\eql@totalrows@
1129 \eql@tagpos@prevrow@\z@
1130 \eql@tagpos@headroom@\z@
1131 \eql@tagpos@footroom@\z@
1132 \eql@dimensions@for{%
1133 \ifnum\eql@tagpos@row@=\eql@totalrows@
1134 \eql@tagpos@headroom@\eql@line@interline@
1135 \eql@tagpos@target@\dimexpr\eql@line@interline@
1136 +\eql@line@height@-\dimen@-\eql@tagheight@block@\relax
1137 \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1138 \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1139 \advance\dimen@\eql@tagpos@target@
1140 \eql@tagpos@target@\z@
1141 \fi
1142 \ifdim\eql@tagpos@target@>%
1143 \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1144 \eql@tagpos@row@\eql@row@
1145 \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1146 \fi
1147 \advance\dimen@-\dimexpr\eql@line@interline@
1148 +\eql@line@depth@+\eql@line@height@\relax
1149 \fi
1150 \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1151 \eql@tagpos@footroom@\eql@line@interline@
1152 \fi
1153 }%
1154 }

```

**TODO:** describe

```

1155 \def\eql@tagpos@print@line@eval{%
1156 \ifdefined\eql@tagpos@continuous
1157 \eql@tagpos@print@line@eval@continuous
1158 \else
1159 \eql@tagpos@print@line@eval@discrete
1160 \fi
1161 }

```

**TODO:** describe

```

1162 \def\eql@tagpos@print@line@eval@continuous{%
1163 \if@eqnsw
1164 \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1165 \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1166 \eql@tagpos@plain@\eql@tagpos@target@
1167 \@tempdima\dimexpr\eql@line@height@+\eql@tagpos@headroom@
1168 -\ht\eql@tagbox@\relax

```

```

1169 \@tempdimb\dimexpr-\eql@line@depth@-\eql@tagpos@footroom@
1170 +\dp\eql@tagbox@\relax
1171 \ifnum\eql@row@=\@ne
1172 \@tempdima.5\maxdimen
1173 \fi
1174 \ifnum\eql@row@=\eql@totalrows@
1175 \@tempdimb-.5\maxdimen
1176 \fi
1177 \ifdim\eql@tagpos@plain@>\@tempdima
1178 \ifdim\eql@tagpos@plain@>\@tempdimb
1179 \ifdim\@tempdima>\@tempdimb
1180 \eql@tagpos@plain@\@tempdima
1181 \else
1182 \eql@tagpos@plain@\@tempdimb
1183 \fi
1184 \fi
1185 \else
1186 \ifdim\eql@tagpos@plain@<\@tempdimb
1187 \ifdim\@tempdima>\@tempdimb
1188 \eql@tagpos@plain@\@tempdimb
1189 \else
1190 \eql@tagpos@plain@\@tempdima
1191 \fi
1192 \fi
1193 \fi
1194 \ifnum\eql@tagpos@prevrow@>\z@
1195 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1196 \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1197 \eql@tagpos@raised@\eql@tagpos@plain@
1198 \let\eql@tagpos@reserve\eql@false
1199 \fi
1200 \else
1201 \ifdim\eql@tagpos@target@>%
1202 \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1203 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1204 \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1205 \eql@tagpos@raised@\eql@tagpos@plain@
1206 \let\eql@tagpos@reserve\eql@false
1207 \fi
1208 \else
1209 \eql@tagpos@raised@\dimexpr-\eql@line@depth@
1210 -\ht\eql@tagbox@\relax
1211 \ifdim\eql@tagpos@raised@<\eql@tagpos@plain@\else
1212 \eql@tagpos@raised@\eql@tagpos@plain@
1213 \let\eql@tagpos@reserve\eql@false
1214 \fi
1215 \fi
1216 \fi
1217 \else
1218 \ifnum\eql@tagpos@prevrow@=\eql@row@
1219 \eql@tagwidth@\eql@tagwidth@block@
1220 \else
1221 \let\eql@tagpos@reserve\eql@false
1222 \fi
1223 \fi
1224 }

```

**TODO:** describe

```

1225 \def\eq@tagpos@print@line@eval@discrete{%
1226   \if@eqnsw
1227     \ht\eq@tagbox@dimexpr\ht\eq@tagbox@-\eq@tagpos@smashup@relax
1228     \dp\eq@tagbox@dimexpr\dp\eq@tagbox@-\eq@tagpos@smashdown@relax
1229     \eq@tagpos@plain@eq@tagpos@shift@
1230     \eq@tagpos@headroom@z@
1231     \eq@tagpos@footroom@z@
1232     \ifdim\eq@tagpos@shift@>%
1233       \ifdefined\eq@tagsleft-1sprelaxelsez@fi
1234       \eq@tagpos@raised@dimexpr\eq@line@height@+\dp\eq@tagbox@relax
1235     else
1236       \eq@tagpos@raised@dimexpr-\eq@line@depth@-\ht\eq@tagbox@relax
1237     fi
1238   \else
1239     \let\eq@tagpos@reserve\eq@false
1240   fi
1241 }

```

**TODO:** describe

```

1242 \def\eq@tagpos@print@line@end{%
1243   \ifdefined\eq@tagpos@continuous
1244     \ifnum\eq@tagpos@prevrow=\eq@row@
1245       \ifdefined\eq@tagpos@reserve
1246         \global\eq@appendexpand\eq@tags@container@block{%
1247           \advance\eq@tagpos@headroom@the\dimexpr\eq@line@height@
1248             +\eq@line@depth@relax}%
1249         \eq@displaybreak@star\M
1250       fi
1251     fi
1252   fi
1253 }

```

## E.6 Component Display

Showkeys Integration. **TODO:** describe

```

1254 \let\eq@SK@loaded\eq@false
1255 \let\eq@SK@label\@gobble
1256 \let\eq@SK@clearlabel\@empty
1257 \let\eq@SK@setlabel\@gobble
1258 \let\eq@SK@printlabel@right\@empty
1259 \let\eq@SK@printlabel@left\@empty
1260 \let\eq@SK@printlabel@line\@empty
1261 \def\eq@label@clean{\eq@label@org}
1262 \AddToHook{package/showkeys/after}{
1263   \let\eq@SK@loaded\eq@true
1264   \def\eq@SK@label#1{\SK@SK@label#1}
1265   \def\eq@SK@clearlabel{\let\eq@SK@lab\relax}
1266   \eq@SK@clearlabel
1267   \def\eq@SK@label#1>#2\SK@{%
1268     \def\eq@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1269   }
1270   \def\eq@SK@setlabel#1{\SK@\eq@SK@label#1}
1271   \def\eq@SK@printlabel@right{%
1272     \ifx\eq@SK@lab\relaxelse
1273       \rlap{\kern\marginparsep\eq@SK@lab}%
1274     \eq@SK@clearlabel

```

```

1275 \fi
1276 }
1277 \def\eq@SK@printlabel@left{%
1278 \ifx\eq@SK@lab\relax\else
1279 \llap{\eq@SK@lab\kern\marginparsep}%
1280 \eq@SK@clearlabel
1281 \fi
1282 }
1283 \def\eq@SK@printlabel@line{%
1284 \ifx\eq@SK@lab\relax\else
1285 \dimen@ \prevdepth
1286 \nointerlineskip
1287 \ifdefined\eq@tags@left
1288 \llap{%
1289 \eq@SK@lab
1290 \kern\marginparsep
1291 }%
1292 \eq@SK@clearlabel
1293 \else
1294 \rlap{%
1295 \dimen@ \displaywidth
1296 \advance\dimen@ \marginparsep
1297 \kern\dimen@
1298 \eq@SK@lab
1299 }%
1300 \fi
1301 \eq@SK@clearlabel
1302 \prevdepth\dimen@
1303 \fi
1304 }
1305 \let\eq@label@org\label
1306 \def\eq@label@clean{\let\SK@\@gobbletwo\eq@label@org}
1307 }

```

## Labels.

`\eq@composetag@label` **TODO:** describe

```

1308 \def\eq@composetag@label{%
1309 \eq@SK@clearlabel
1310 \ifdefined\eq@tags@label
1311 \eq@SK@setlabel\eq@tags@label
1312 \ifdefined\eq@tags@name
1313 \let\@currentlabelname\eq@tags@name
1314 \else
1315 \let\@currentlabelname\eq@tags@name@generic
1316 \fi
1317 \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1318 \fi
1319 }

```

**TODO:** describe

```

1320 \def\eq@tags@printsubeq@label{%
1321 \eq@tags@container@parent
1322 \ifdefined\eq@tags@label
1323 \eq@tags@makeblockanchor
1324 \eq@SK@setlabel\eq@tags@label
1325 \begingroup

```

```

1326     \def\@currentcounter{equation}%
1327     \eq@tags@anchor
1328     \let\@currentlabelname\eq@tags@name@generic
1329     \protected@edef\@currentlabel{\p@equation\theparentequation}%
1330     \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1331     \endgroup
1332     \eq@SK@printlabel@line
1333 \fi
1334 }

```

**Hyperref Anchors.** **TODO:** describe

```

1335 \let\eq@Hy@anchor\@gobble
1336 \AddToHook{package/hyperref/after}{
1337   \def\eq@Hy@anchor#1{%
1338     \Hy@raisedlink{\hyper@anchor{#1}}%
1339   }%
1340 }

```

**TODO:** describe

```

1341 \def\eq@tags@makeblockanchor{%
1342   \eq@tags@glabel@step
1343   \eq@Hy@anchor\eq@tags@glabel
1344   \edef\eq@tags@anchor{%
1345     \def\noexpand\thepage{\thepage}%
1346     \def\noexpand\@currentHref{\eq@tags@glabel}%
1347   }%
1348 }

```

**TODO:** describe

ql@composetag@anchor

```

1349 \def\eq@composetag@anchor{%
1350   \ifdefined\eq@tags@tag
1351     \def\@currentcounter{equation}%
1352     \ifdefined\eq@tags@ref
1353       \let\@currentlabel\eq@tags@ref
1354     \else
1355       \protected@edef\@currentlabel{\p@equation\eq@tags@tag}%
1356     \fi
1357     \eq@tags@glabel@step
1358     \edef\@currentHref{\eq@tags@glabel}%
1359     \eq@Hy@anchor\@currentHref
1360   \else
1361     \refstepcounter{equation}%
1362     \protected@edef\eq@tags@tag{\theequation}%
1363   \fi
1364   \eq@tags@anchor
1365 }

```

**Tag Layout.** **TODO:** describe

```

1366 \def\eq@tags@taglayout@set@direct#1{%
1367   \def\eq@tags@taglayout##1{#1}%
1368 }
1369 \def\eq@tags@taglayout@set#1{%

```



```

1370 \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1371 }

```

**TODO:** describe

```

1372 \def\eql@tags@tagform@set@direct#1{%
1373   \def\eql@tags@tagform##1{#1}%
1374 }
1375 \def\eql@tags@tagform@set#1#2#3{%
1376   \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1377 }

1378 \eql@tags@taglayout@set{#1}
1379 \eql@tags@tagform@set({#1})
1380 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1381 \protected\def\tagform{\eql@tags@tagform}
1382 \protected\def\tagbox{\eql@tags@taglayout}
1383 \protected\def\tagboxed{\eql@tags@tagcompose}

```

`\eqref` `amsmath` defines the macro `\eqref` to refer to equation labels in a proper format. We provide it for completeness:

```

1384 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

`\eql@composetag@tag` **TODO:** describe

```

1385 \def\eql@composetag@tag{%
1386   \eql@tagging@tagbegin
1387   \eql@tags@frame@cmd{%
1388     \eql@tags@taglayout{%
1389       \eql@tags@tagform\eql@tags@tag
1390       \eql@tagging@tagsave
1391     }%
1392   }%
1393   \eql@tagging@tagend
1394 }

```

## E.7 Tag Composition

**TODO:** describe

```

1395 \def\eql@composetag@measure{%
1396   \ifdefined\eql@tags@tag\else
1397     \stepcounter{equation}%
1398     \let\eql@tags@tag\theequation
1399   \fi
1400   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1401   \ifdefined\eql@numbering@multi
1402     \global\let\eql@tags@container\eql@tags@container@clear
1403   \fi
1404 }

```

**TODO:** describe

```

1405 \def\eql@composetag@print{%
1406   \eql@composetag@anchor
1407   \eql@composetag@label
1408   \ifdefined\eql@tagsleft

```

```

1409 \eq@SK@printlabel@left
1410 \eq@composetag@tag
1411 \else
1412 \eq@composetag@tag
1413 \eq@SK@printlabel@right
1414 \fi
1415 \global\let\eq@tags@container\eq@tags@container@clear
1416 }

```

**TODO:** describe

**TODO:** one might still compare width to zero and pretend the tag is absent??

```

1417 \def\eq@tagbox@make#1{%
1418 \setbox\eq@tagbox@\hbox{\eq@strut@tag\@lign#1}%
1419 \eq@tagwidth@\wd\eq@tagbox@
1420 \ifdim\eq@tagwidth@<\eq@tagwidthmin@
1421 \eq@tagwidth@\eq@tagwidthmin@
1422 \fi
1423 \advance\eq@tagwidth@\eq@tagsepmin@
1424 }

```

**TODO:** describe

```

1425 \def\eq@tagbox@print@adjustheadroom{%
1426 \dimen@\dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1427 \ifdim\dimen@>\z@
1428 \ifdim\dimen@>\eq@tagpos@headroom@
1429 \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@headroom@\relax
1430 \else
1431 \ht\eq@tagbox@\dimexpr\eq@line@height@-\eq@tagpos@current@\relax
1432 \fi
1433 \fi
1434 }

```

**TODO:** describe

```

1435 \def\eq@tagbox@print@adjustfootroom{%
1436 \dimen@\dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax
1437 \ifdim\dimen@>\z@
1438 \ifdim\dimen@>\eq@tagpos@footroom@
1439 \dp\eq@tagbox@\dimexpr\dp\eq@tagbox@-\eq@tagpos@footroom@\relax
1440 \else
1441 \dp\eq@tagbox@\dimexpr\eq@line@depth@+\eq@tagpos@current@\relax
1442 \fi
1443 \fi
1444 }

```

**TODO:** describe

```

1445 \def\eq@tagbox@print@extendabove{%
1446 \dimen@\dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1447 \ifdim\dimen@>\z@
1448 \global\eq@appendexpand\eq@display@container{%
1449 \eq@display@aboveextend@the\dimen@\relax}%
1450 \fi
1451 }

```

**TODO:** describe

```

1452 \def\eq@tagbox@print@extendbelow{%
1453 \dimen@\dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax

```

```

1454 \ifdim\dimen@>\z@
1455   \global\eql@appendexpand\eql@display@container{%
1456     \eql@display@belowextend@the\dimexpr\dimen@relax}%
1457 \fi
1458 }

```

**TODO:** describe

```

1459 \def\eql@tagbox@print@prepare{%
1460   \ifdefined\eql@tagpos@reserve
1461     \eql@tagpos@current@\eql@tagpos@plain@
1462   \else
1463     \eql@tagpos@current@\eql@tagpos@raised@
1464   \fi
1465   \ifdim\eql@tagpos@headroom@>\z@
1466     \eql@tagbox@print@adjustheadroom
1467   \fi
1468   \ifdim\eql@tagpos@footroom@>\z@
1469     \eql@tagbox@print@adjustfootroom
1470   \fi
1471   \ifnum\eql@row@=\@ne
1472     \eql@tagbox@print@extendabove
1473   \fi
1474   \ifnum\eql@row@=\eql@totalrows@
1475     \eql@tagbox@print@extendbelow
1476   \fi
1477 }

```

**TODO:** describe

```

1478 \def\eql@tagbox@print@tagsright{%
1479   \eql@tagbox@print@prepare
1480   \kern-\wd\eql@tagbox@
1481   \raise\eql@tagpos@current@\box\eql@tagbox@
1482 }

```

**TODO:** describe

```

1483 \def\eql@tagbox@print@tagsleft{%
1484   \eql@display@firstavail@set\z@
1485   \eql@tagbox@print@prepare
1486   \wd\eql@tagbox@\z@
1487   \raise\eql@tagpos@current@\box\eql@tagbox@
1488 }

```

$\eql@tagbox@print@cell$

```

1489 \def\eql@tagbox@print@cell{%
1490   \eql@tagging@tagaddbox
1491   \ifdefined\eql@tagsleft
1492     \ifdefined\eql@tagpos@reserve
1493       \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@relax
1494         \let\eql@tagpos@reserve\eql@false
1495       \fi
1496     \fi
1497     \if@eqnsw
1498       \eql@tagbox@print@tagsleft
1499     \fi
1500     \kern\displaywidth
1501   \else
1502     \kern\displaywidth

```

```

1503 \ifdefined\eq\tagpos@reserve
1504 \ifdim\eq\tagwidth@>%
1505 \dimexpr\displaywidth-\eq\line@width@+\eq\tagfuzz@\relax
1506 \let\eq\tagpos@reserve\eq\false
1507 \fi
1508 \fi
1509 \if@eqnsw
1510 \eq\tagbox@print@tagsright
1511 \fi
1512 \fi
1513 }

```

## F Subequation Numbering

We replicate the `amsmath` functionality to number a block of equations with a common number and a sub-numbering.

### F.1 Definitions

`parentequation (counter)` We define a counter to store the main equation number while in subequation mode. It makes sense to share this definition with `amsmath` as `parentequation`, and we need to undefine it when `amsmath` is loaded at a later stage:

```

1514 \eq\amsmath@undefine\c@parentequation
1515 \eq\amsmath@undefine\theparentequation
1516 \ifdefined\c@parentequation\else
1517 \newcounter{parentequation}
1518 \fi

```

`subequations@template` We store a template which will be installed as `\theequation` in subequations mode: **TODO:** need to protect something?!

```

1519 \def\eq\subequations@template{\theparentequation\alph{equation}}

```

`@subequations@active` A boolean register which tells whether subequations are in use and thus must not be invoked again:

```

1520 \let\eq\subequations@active\eq\false

```

`\eq\subequations@init` Low-level initialise the subequations mode. Store the equation counter in `\eq\subequations@restorecounter` for the case that no equation numbers will be used. Step the equation counter, copy it to `parentequation` and initialise `\theparentequation` (and its `hyperref` counterpart) with the expanded current value of `\theequation`; fill with tag data instead if a tag has been specified. Reset the equation counter and use the template for `\theequation`:

```

1521 \def\eq\subequations@init{%
1522 \edef\eq\subequations@restorecounter{%
1523 \global\c@equation\the\c@equation\relax}%
1524 \eq\tags@container@block
1525 \ifdefined\eq\tags@tag
1526 \eq\tags@glabel@step
1527 \protected@edef\theparentequation{\eq\tags@glabel}%
1528 \protected@edef\theparentequation{\eq\tags@tag}%
1529 \else

```

```

1530 \advance\c@equation\@ne
1531 \protected@edef\theparentequation{\theequation}%
1532 \ifdefined\theHequation
1533 \protected@edef\theHparentequation{\theHequation}%
1534 \fi
1535 \fi
1536 \global\c@parentequation\c@equation
1537 \global\c@equation\z@
1538 \let\theequation\eq@subequations@template
1539 \def\theHequation{\theHparentequation.\arabic{equation}}%
1540 }

```

**1@subequations@close** Low-level close the subequations mode. If no number has been used, return to the original equation counter, otherwise use the value stored in `parentequation`. Note that we cannot use `\setcounter` here because the `calc` version would involve actions which are not allowed after `\halign` within a display equation:

```

1541 \def\eq@subequations@close{%
1542 \ifnum\c@equation=\z@
1543 \eq@subequations@restorecounter
1544 \else
1545 \global\c@equation\c@parentequation
1546 \fi
1547 }

```

## F.2 Environment

**1@subequations@start** Start the subequations environment with optional parameters in #1. Enter subequations mode and set an anchor for subsequent `\label`'s. Manually print the `showkeys` tag:  
**TODO:** join with other similar anchor routines `\eq@tags@printslabel`

```

1548 \def\eq@subequations@start{%
1549 \let\eq@tags@container@block\eq@tags@container@clear
1550 \eq@nextopt@process{subequations}%
1551 \eq@subequations@init
1552 \eq@tags@glabel@step
1553 \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1554 \eq@Hy@anchor\eq@subequations@currentHref
1555 \edef\eq@subequations@thepage{\thepage}%
1556 \def\@currentcounter{equation}%
1557 \let\@currentHref\eq@subequations@currentHref
1558 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1559 \eq@tags@container@block
1560 \ifdefined\eq@tags@name
1561 \let\@currentlabelname\eq@tags@name
1562 \else
1563 \let\@currentlabelname\eq@tags@name@generic
1564 \fi
1565 \let\eq@subequations@active\eq@true
1566 \ifdefined\eq@tags@label
1567 \eq@SK@label\eq@tags@label
1568 \fi
1569 \ignorespaces
1570 }

```

**eq@subequations@end** End the subequations environment. Issue the label if one has been specified and an equation number has actually been used. Then close subequations mode:

```

1571 \def\eq@subequations@end{%
1572   \ifnum\c@equation>\z@
1573     \eq@tags@container@block
1574     \ifdefined\eq@tags@label
1575       \begingroup
1576         \def\@currentcounter{equation}%
1577         \let\thepage\eq@subequations@thepage
1578         \let\@currentHref\eq@subequations@currentHref
1579 % \TODO how about tag* ?! also within equations!
1580         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1581         \ifdefined\eq@tags@name
1582           \let\@currentlabelname\eq@tags@name
1583         \else
1584           \let\@currentlabelname\eq@tags@name@generic
1585         \fi
1586         \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1587       \endgroup
1588     \fi
1589   \fi
1590   \eq@subequations@close
1591   \ignorespacesafterend
1592 }

```

`subequations` (*env.*) The subequations environment tests for optional parameters and passes on to the start and end routines:

```

1593 \newenvironment{eq@subequations}{%
1594 (dev)\eq@dev@enterenv
1595   \eq@subequations@testall\eq@subequations@start%
1596 }{%
1597   \eq@subequations@end
1598 (dev)\eq@dev@leaveenv
1599 }

```

**TODO:** describe

```

1600 \def\eq@subequations@testall{\eq@parseopt\eq@subequations@parseopt}
1601 \def\eq@subequations@parseopt{%
1602   \ifx\eq@parseopt@token[%]
1603     \let\eq@parseopt@next\eq@parseopt@opt
1604   \fi
1605   \ifx\eq@parseopt@token\eq@atxi
1606     \let\eq@parseopt@next\eq@parseopt@label
1607   \fi
1608   \ifx\eq@parseopt@token\eq@atxii
1609     \let\eq@parseopt@next\eq@parseopt@label
1610   \fi
1611   \ifx\eq@parseopt@token\label
1612     \let\eq@parseopt@next\eq@parseopt@end
1613   \fi
1614 }

```

### F.3 Subequation Scheme

**TODO:** describe

```

1615 \def\eq@numbering@subeq@init{%
1616   \let\eq@save@theequation\theequation

```

```

1617 \let\eql@save@theHequation\theHequation
1618 \eql@subequations@init
1619 \let\eql@tags@container@parent\eql@tags@container@block
1620 \let\eql@tags@container@block\eql@tags@container@clear
1621 }

```

**TODO:** describe

```

1622 \def\eql@numbering@subeq@test{%
1623   \ifnum\eql@tagrows@<\tw@
1624     \let\eql@tags@container@block\eql@tags@container@parent
1625     \let\eql@numbering@subeq@use\eql@false
1626     \let\theequation\eql@save@theequation
1627     \let\theHequation\eql@save@theHequation
1628     \eql@subequations@restorecounter
1629   \fi
1630 }

```

**TODO:** describe

```

1631 % \TODO note must not use setcounter here (when calc is loaded)
1632 \def\eql@numbering@subeq@close{%
1633   \eql@subequations@close
1634 }

```

## G Display Equations Support

**TODO:** describe

```

1635 \let\eql@display@injectbefore\@undefined
1636 \let\eql@display@injectafter\@undefined
1637 \let\eql@interline@container\@undefined
1638 \def\eql@interline@container@clear{%
1639   \eql@displaybreak@pen@\@MM
1640   \eql@vspaceskip@\z@skip
1641 }

```

### G.1 Display Breaks

**TODO:** describe

erdisplaylinepenalty

```

1642 \interdisplaylinepenalty\@M

```

\eql@getdsp@pen **TODO:** isn't this the opposite order than \@getpen?!

```

1643 \def\eql@getdsp@pen#1{%
1644   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1645 }

```

**TODO:** allow a displaybreak before equations

```

1646 \protected\def\eql@displaybreak@default{%
1647   \eql@warning{Invalid use of \string\displaybreak}{}%
1648   \eql@teststaropt@loose\@gobble\eql@gobbleopt{}}
1649 \eql@amsmath@after{\let\eql@displaybreak@default\displaybreak}
1650 \eql@amsmath@let\displaybreak\eql@displaybreak@default

```

```

1651 \newcount\eqldisplaybreak@pen@
1652 \newcount\eqldisplaybreak@prepen@
1653 \newcount\eqldisplaybreak@postpen@

```

**TODO:** describe

```

1654 \protected\def\eqldisplaybreak{%
1655   \relax
1656   \eql@ampprotecttwo\eql@teststaroropt@tight
1657   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1658 }

1659 \def\eqldisplaybreak@star#1{%
1660   \global\eql@appendexpand\eql@interline@container{%
1661     \eqldisplaybreak@pen@\the\numexpr#1\relax\relax}%
1662 }

1663 \def\eqldisplaybreak@level[#1]{%
1664   \ifnum#1<\z@
1665     \global\eql@append\eql@interline@container{\eqldisplaybreak@pen@\@MM}%
1666   \else
1667     \global\eql@appendexpand\eql@interline@container{%
1668       \eqldisplaybreak@pen@-\@getpen{#1}\relax}%
1669   \fi
1670 }

```

**TODO:** describe

```

1671 \def\eqldisplaybreak@pre#1{%
1672   \ifnum#1<\z@
1673     \eqldisplaybreak@prepen@\@MM
1674   \else
1675     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1676   \fi
1677 }

```

**TODO:** describe

```

1678 \def\eqldisplaybreak@post#1{%
1679   \ifnum#1<\z@
1680     \eqldisplaybreak@postpen@\@MM
1681   \else
1682     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1683   \fi
1684 }

```

**TODO:** describe

```

1685 \def\eqldisplaybreak@inter#1{%
1686   \ifnum#1<\z@
1687     \interdisplaylinepenalty\@M
1688   \else
1689     \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax
1690   \fi
1691 }

```

## G.2 Explicit Vertical Space

**TODO:** describe



`eql@vspaceskip@ (skip)`

```
1692 \newskip\eql@vspaceskip@

1693 \let\eql@vspace@org\vspace
1694 \def\eql@vspace{%
1695   \ifvmode
1696     \expandafter\eql@vspace@immediate
1697   \else
1698     \expandafter\eql@vspace@line
1699   \fi
1700 }
```

**TODO:** `\eql@vspace@addfixedafter` on last line has no effect. should apply outside environment

```
1701 \def\eql@vspace@line{%
1702   \eql@ifstar@loose\eql@vspace@addfixedbefore\eql@vspace@add
1703 }
1704 \def\eql@vspace@add#1{%
1705   \global\eql@appendexpand\eql@interline@container{%
1706     \advance\eql@vspaceskip@\the\glueexpr#1\relax\relax}%
1707 \def\eql@vspace@addfixedbefore#1{%
1708   \global\eql@appendexpand\eql@interline@container{%
1709     \noexpand\eql@append\noexpand\eql@display@injectbefore{%
1710       \skip@\the\glueexpr#1\relax\relax
1711       \penalty\@M
1712       \vskip\skip@
1713       \global\advance\eql@line@interline@\skip@
1714     }%
1715   }%
1716 }
1717 \def\eql@vspace@addfixedafter#1{%
1718   \global\eql@appendexpand\eql@interline@container{%
1719     \noexpand\eql@append\noexpand\eql@display@injectafter{%
1720       \dimen@\prevdepth
1721       \hrule\@height\z@
1722       \skip@\the\glueexpr#1\relax\relax
1723       \penalty\@M
1724       \vskip\skip@
1725       \global\advance\eql@line@interline@\skip@
1726       \prevdepth\dimen@
1727     }%
1728   }%
1729 }
```

**TODO:** careful to not expand `\eql@display@container` after measure

```
1730 \def\eql@vspace@immediate{%
1731   \noalign\bgroup
1732     \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
1733 }
1734 \def\eql@vspace@fixed#1{%
1735   \skip@\glueexpr#1\relax
1736   \ifnum\eql@row@=\@ne
1737     \global\eql@appendexpand\eql@display@container{%
1738       \advance\eql@abovespace@\the\skip@\relax}%
1739   \else\ifnum\eql@row@>\eql@totalrows@
1740     \global\eql@appendexpand\eql@display@container{%
1741       \advance\eql@belowspace@\the\skip@\relax}%
1742   \fi
1743 }
```

```

1742     \else
1743     \dimen@\prevdepth
1744     \hrule\@height\z@
1745     \penalty\@M
1746     \vskip\skip@
1747     \global\advance\eql@line@interline@\skip@
1748     \prevdepth\dimen@
1749     \fi\fi
1750 \egroup
1751 }
1752 \def\eql@vspace@discardable#1{%
1753     \skip@\glueexpr#1\relax
1754     \ifnum\eql@row@=\@one
1755         \global\eql@appendexpand\eql@display@container{%
1756             \advance\eql@abovespace@\the\skip@\relax}%
1757     \else\ifnum\eql@row@>\eql@totalrows@
1758         \global\eql@appendexpand\eql@display@container{%
1759             \advance\eql@belowspace@\the\skip@\relax}%
1760     \else
1761         \vskip\skip@
1762         \global\advance\eql@line@interline@\skip@
1763     \fi\fi
1764 \egroup
1765 }

```

### G.3 Default Vertical Spacing

**TODO:** describe

`\eql@strut` Next follows a special internal strut which is supposed to match the height and the depth of a normal `\strut` minus `\normallineskiplimit` according to M. Spivak.

```

1766 \newbox\eql@strutbox@
1767 \def\eql@strut@depth{.3}
1768 \def\eql@strut{\copy\eql@strutbox@}
1769 \let\eql@strut@cell\eql@strut
1770 \let\eql@strut@tag\eql@strut
1771 \def\eql@strut@make{%
1772     \setbox\eql@strutbox@\hbox{%
1773         \@tempdimb\dimexpr
1774             \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1775         \@tempdima\dimexpr
1776             \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
1777         \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1778     }
1779 }
1780 \AtBeginDocument{\eql@strut@make}

```

**TODO:** describe

```

1781 \def\eql@spread@set{%
1782     \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
1783     +\normalbaselineskip-\baselineskip\relax
1784     \ifdim\eql@spread@>\z@
1785         \openup\eql@spread@
1786         \ifdefined\spread@equation
1787             \let\spread@equation\@empty
1788         \fi

```

```

1789 \fi
1790 }

```

## G.4 Entry and Exit

**TODO:** describe

```

1791 \let\eql@beamerbasecolor@fix\@empty
1792 \AddToHook{package/beamerbasecolor/after}{%
1793   \def\eql@beamerbasecolor@fix{%
1794     \donotcolorouterdisplaymaths
1795     \donotcoloroutermaths
1796     \beamer@setdisplaymathcolor
1797   }%
1798 }

```

`\eql@abovespace@` (*skip*)

`\eql@belowspace@` (*skip*)

```

1799 \newskip\eql@abovespace@
1800 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

1801 \def\eql@display@enter{%
1802   \if@noskipsec\leavevmode\par\fi
1803   \ifvmode
1804     \eql@prevdepth@\prevdepth
1805     \nointerlineskip
1806     \noindent
1807   \else
1808     \eql@prevdepth@\maxdimen
1809   \fi
1810   \eql@beamerbasecolor@fix
1811 }

```

`\eql@display@adjust`

```

1812 \def\eql@display@adjust{%
1813   \ifdefined\eql@display@linewidth
1814     \displaywidth\glueexpr\eql@display@linewidth\relax
1815     \advance\displaywidth-\displayindent
1816   \fi
1817   \ifdefined\eql@display@marginleft
1818     \advance\displaywidth\displayindent
1819     \displayindent\glueexpr\eql@display@marginleft\relax
1820     \advance\displaywidth-\displayindent
1821   \fi
1822   \ifdefined\eql@display@marginright
1823     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1824   \fi
1825   \ifdim\displaywidth<\z@
1826     \displaywidth\z@
1827   \fi
1828 }

```

`\eql@display@init`

```

1829 \def\eqldisplay@init{%
1830   \let\displaybreak\eqldisplaybreak
1831   \let\eqlvspace@org\vspace
1832   \let\vspace\eqlvspace
1833   \let\eqncontrol\eql@control
1834   \let\eqldisplay@injectbefore\@empty
1835   \let\eqldisplay@injectafter\@empty
1836   \eql@spread@set
1837   \eql@strut@make
1838   \let\eql@frame@cmd\@undefined
1839 }

```

\eqldisplay@print

```

1840 \def\eqldisplay@print{%
1841   \let\eqldisplay@container\@empty
1842   \eqldisplay@firstavail@z@
1843   \eqldisplay@aboveextend@z@
1844   \eqldisplay@belowextend@z@
1845   \global\let\eql@interline@container\eql@interline@container@clear
1846 }

```

@display@halign@init **TODO:** describe

```

1847 \def\eqldisplay@halign@init#1{%
1848   \eql@row@z@
1849   \eql@prevgraf@\prevgraf
1850   \everycr{\noalign{%
1851     \global\advance\eql@row@\@ne
1852     \prevgraf\numexpr\prevgraf+\@ne\relax
1853     #1%
1854   }}%
1855 }

```

**TODO:** how about penalty here? not for entry into display

```

1856 \def\eqldisplay@halign@start{%
1857   \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1858   \ifdim\eql@prevdepth=\maxdimen\else
1859     \prevdepth\eql@prevdepth@
1860   \fi
1861   \ifdim\prevdepth=-\@m\p@else
1862     \ifdefined\eqldisplay@height
1863       \skip@\baselineskip
1864       \advance\skip@-\glueexpr\eqldisplay@height\relax
1865       \advance\skip@-\prevdepth\relax
1866       \ifdim\skip@<\lineskiplimit
1867         \skip@\lineskip
1868       \fi
1869       \advance\skip@-\eql@spread@\relax
1870       \vskip\skip@
1871       \nointerlineskip
1872     \else
1873       \vskip-\eql@spread@\relax
1874     \fi
1875   \fi
1876 }

```

**TODO:** describe

```

1877 \def\eqldisplay@vspace{%
1878   \advance\abovedisplayskip\eql@abovespace@
1879   \advance\belowdisplayskip\eql@belowspace@
1880 }

```

**TODO:** describe

```

1881 \def\eqldisplay@vspace@native{%
1882   \advance\abovedisplayskip\eql@abovespace@
1883   \advance\belowdisplayskip\eql@belowspace@
1884   \advance\abovedisplayshortskip\eql@abovespace@
1885   \advance\belowdisplayshortskip\eql@belowspace@
1886 }

```

**TODO:** describe

```

1887 \def\eqldisplay@penalty{%
1888   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1889     \postdisplaypenalty\eqldisplaybreak@postpen@
1890   \fi
1891   \ifnum\eqldisplaybreak@pen@=\@MM\else
1892     \postdisplaypenalty\eqldisplaybreak@pen@
1893   \fi
1894   \ifnum\eqldisplaybreak@prepen@=\@MM\else
1895     \predisplayskip\eqldisplaybreak@prepen@
1896   \fi
1897 }

```

**TODO:** describe **TODO:** issue: `\vspace*{0pt}` has some effect if page is broken here

```

1898 \def\eqldisplay@halign@end{%
1899   \eql@interline@container
1900   \eqldisplay@injectbefore
1901   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
1902   \ifdefined\eqldisplay@depth
1903     \prevdepth\glueexpr\eqldisplay@depth\relax
1904   \fi
1905 }

```

`\eqldisplay@close` **TODO:** there seems to be an offset of 1em in `\predisplaysize` towards actual content, nice.  
**TODO:** must not use `\setlength` or `\setcounter` when `\calc` is loaded **TODO:** do we actually need penalty adjustments in case of paragraphs above or below?

```

1906 \def\eqldisplay@close{%
1907   \eqldisplay@container
1908   \ifdim\eqldisplay@firstavail@<\z@
1909     \eqldisplay@firstavail@\z@
1910   \fi
1911   \eql@skip@mode@leave@\z@
1912   \ifdim\eql@prevdepth@=\maxdimen
1913     \ifdim\predisplaysize=-\maxdimen
1914       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1915       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1916     \else
1917       \eql@skip@mode@above@\z@
1918       \eql@skip@mode@below@\z@
1919       \advance\eqldisplay@firstavail@\displayindent
1920       \ifdim\eqldisplay@firstavail@>\predisplaysize
1921         \ifcase\eql@skip@mode@short\relax
1922         \or

```

```

1923         \eqL@skip@mode@above@ \@ne
1924     \or
1925         \eqL@skip@mode@above@ \@ne
1926         \ifnum\eqL@totalrows@=\@ne
1927             \eqL@skip@mode@below@ \@ne
1928         \fi
1929     \or
1930         \eqL@skip@mode@above@ \@ne
1931         \eqL@skip@mode@below@ \@ne
1932     \fi
1933 \fi
1934 \fi
1935 \else
1936     \ifdim\eqL@prevdepth@=-\@m\p@
1937         \eqL@skip@mode@above@ \eqL@skip@mode@top@above\relax
1938         \eqL@skip@mode@below@ \eqL@skip@mode@top@below\relax
1939     \else
1940         \eqL@skip@mode@above@ \eqL@skip@mode@par@above\relax
1941         \eqL@skip@mode@below@ \eqL@skip@mode@par@below\relax
1942     \fi
1943 \fi
1944 \ifcase\eqL@skip@mode@above@
1945 \or\or\or
1946     \predisplaypenalty\z@
1947 \or
1948     \predisplaypenalty\z@
1949 \fi
1950 \ifcase\eqL@skip@mode@below@
1951 \or\or\or
1952     \eqL@skip@mode@leave@ \@ne
1953 \or
1954     \eqL@skip@mode@leave@ \tw@
1955 \fi
1956 \ifdefined\eqL@skip@force@above
1957     \eqL@skip@mode@above@ \eqL@skip@force@above\relax
1958 \fi
1959 \ifdefined\eqL@skip@force@below
1960     \eqL@skip@mode@below@ \eqL@skip@force@below\relax
1961 \fi
1962 \ifdefined\eqL@skip@force@leave
1963     \eqL@skip@mode@leave@ \eqL@skip@force@leave\relax
1964 \fi
1965 \ifnum\eqL@skip@mode@leave@>\z@
1966     \postdisplaypenalty\z@
1967 \fi
1968 \ifcase\eqL@skip@mode@above@
1969     \abovedisplayskip\glueexpr\eqL@skip@long@above\relax
1970 \or
1971     \abovedisplayskip\glueexpr\eqL@skip@short@above\relax
1972 \or
1973     \abovedisplayskip\glueexpr\eqL@skip@cont@above\relax
1974 \or
1975     \abovedisplayskip\glueexpr\eqL@skip@par@above\relax
1976 \or
1977     \abovedisplayskip\glueexpr\eqL@skip@top@above\relax
1978 \or
1979     \abovedisplayskip\z@skip
1980 \or

```

```

1981 \abovedisplayskip\glueexpr\eq\@skip@med@above\relax
1982 \or
1983 \abovedisplayskip\glueexpr\eq\@skip@custom@above\relax
1984 \fi
1985 \ifcase\eq\@skip@mode@below@
1986 \belowdisplayskip\glueexpr\eq\@skip@long@below\relax
1987 \or
1988 \belowdisplayskip\glueexpr\eq\@skip@short@below\relax
1989 \or
1990 \belowdisplayskip\glueexpr\eq\@skip@cont@below\relax
1991 \or
1992 \belowdisplayskip\glueexpr\eq\@skip@par@below\relax
1993 \or
1994 \belowdisplayskip\glueexpr\eq\@skip@top@below\relax
1995 \or
1996 \belowdisplayskip\z@skip
1997 \or
1998 \belowdisplayskip\glueexpr\eq\@skip@med@below\relax
1999 \or
2000 \belowdisplayskip\glueexpr\eq\@skip@custom@below\relax
2001 \fi
2002 \global\eq\@skip@mode@leave@\eq\@skip@mode@leave@
2003 \eq\@interline@container
2004 \advance\eq\@belowspace@\eq\@vspaceskip@
2005 \eq\@display@penalty
2006 \eq\@display@vspace
2007 \skip@\glueexpr\eq\@skip@tag@above\relax
2008 \ifdim\skip@>\abovedisplayskip
2009 \skip@\abovedisplayskip
2010 \fi
2011 \advance\abovedisplayskip-\eq\@display@aboveextend@\relax
2012 \ifdim\abovedisplayskip<\skip@
2013 \abovedisplayskip\skip@
2014 \fi
2015 \skip@\glueexpr\eq\@skip@tag@below\relax
2016 \ifdim\skip@>\belowdisplayskip
2017 \skip@\belowdisplayskip
2018 \fi
2019 \ifdim\eq\@display@belowextend@>\z@
2020 \advance\belowdisplayskip-\eq\@display@belowextend@\relax
2021 \ifdim\belowdisplayskip<\skip@
2022 \belowdisplayskip\skip@
2023 \fi
2024 \fi
2025 }

```

**TODO:** describe

```

2026 \def\eq\@display@leave{%
2027 \prevgraf\eq\@prevgraf@
2028 \ifcase\eq\@skip@mode@leave@
2029 \or
2030 \endgraf
2031 \or
2032 \endgraf
2033 \prevdepth-\@m\p@
2034 \fi
2035 }

```

**TODO:** describe

```
2036 \def\eqldisplay@nest{%
2037   \let\displaybreak\eqldisplaybreak@default
2038   \let\intertext\eqlintertext@default
2039   \let\vspace\eqlvspace@org
2040 }
```

**TODO:** describe

```
2041 \def\eqldisplay@restore{%
2042   \let\label\eqllabel@org
2043   \let>tag\eqltag@default
2044   \let\raisetag\eqlraisetag@default
2045   \let\displaybreak\eqldisplaybreak@default
2046   \let\intertext\eqlintertext@default
2047   \let\vspace\eqlvspace@org
2048 }
```

**TODO:** describe

```
2049 \eqlappend\@arrayparboxrestore{%
2050   \eqldisplay@restore
2051   \ifdefined\eql@ampproof@active
2052     \eql@amprevert
2053   \fi
2054   \@displayfalse
2055 }
```

## G.5 Stack

**TODO:** describe **TODO:** for each global variable declare global nature at its definition!

**TODO:** we must be consistent about global variables vs local variables global variables need to be saved at every level where they may be modified (even if modified only locally)

```
2056 \def\eql@stack@enable{%
2057   \let\eql@stack@save@equations\eql@stack@save@equations@
2058   \let\eql@stack@save@box\eql@stack@save@box@
2059 }
```

**TODO:** describe

```
2060 \let\eql@stack@save@equations\eql@stack@enable
2061 \let\eql@stack@save@box\eql@stack@enable
2062 \let\eql@stack@restore\@empty
```

**TODO:** describe

```
2063 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
2064 \def\eql@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

**TODO:** further global variables: global registers: \eql@nextopt, \eql@tags@glabel@ used locally without possibility of change between setting and retrieving:

\eql@prevgraf@, \eql@skip@mode@leave@, \eql@shape@lastrow, \eql@frame@prevcmd

**TODO:** to be reviewed: \eql@intertext@after, \eql@intertext@opt **TODO:** describe

```
2065 \def\eql@stack@save@equations@{%
2066   \let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2067   \let\eql@stack@cell@container\eql@cell@container
2068   \let\eql@stack@tags@container\eql@tags@container
```



```

2069 \let\eql@stack@interline@container\eql@interline@container
2070 \let\eql@stack@block@container\eql@display@container
2071 \let\eql@stack@dimensions@tab\eql@dimensions@tab
2072 \edef\eql@stack@restore{%
2073   \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2074   \eql@stack@save@let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2075   \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2076   \eql@stack@save@let\eql@stack@tags@container\eql@tags@container
2077   \eql@stack@save@let\eql@stack@interline@container\eql@interline@container
2078   \eql@stack@save@let\eql@stack@dimensions@tab\eql@dimensions@tab
2079   \eql@stack@save@let\eql@stack@block@container\eql@display@container
2080   \eql@stack@save@reg\eql@column@
2081   \eql@stack@save@reg\eql@totalcolumns@
2082   \eql@stack@save@reg\eql@line@avail@
2083   \eql@stack@save@reg\eql@line@pos@
2084   \eql@stack@save@reg\eql@line@width@
2085   \eql@stack@save@reg\eql@line@depth@
2086   \eql@stack@save@reg\eql@line@height@
2087   \eql@stack@save@reg\eql@line@prevdepth@
2088   \eql@stack@save@reg\eql@line@interline@
2089   \eql@stack@save@reg\eql@totalheight@
2090   \eql@stack@save@reg\eql@tagwidth@max@
2091   \eql@stack@save@reg\eql@tagpos@row@
2092   \eql@stack@save@reg\eql@row@
2093   \eql@stack@save@reg\eql@tagrows@
2094 }%
2095 }

```

**TODO:** describe

```

2096 \def\eql@stack@save@box{%
2097   \let\eql@stack@cell@container\eql@cell@container
2098   \edef\eql@stack@restore{%
2099     \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2100     \eql@stack@save@reg\eql@row@
2101   }%
2102 }

```

## H Multi-Line Support

**TODO:** describe

### H.1 Measure Support

**TODO:** describe

```

2103 \def\eql@measure@init#1#2{%
2104   \eql@dimensions@reset
2105   \let\eql@display@container\@empty
2106   \eql@numbering@measure@init
2107   \eql@row@\z@
2108   \eql@totalheight@\z@
2109   \eql@totalrows@\@M
2110   \eql@line@prevdepth@-\@m\p@
2111   \eql@line@interline@\z@
2112   \tabskip\z@skip
2113   \everycr{\noalign{%

```

```

2114 \global\advance\eq\row@\@ne
2115 #1%
2116 }%
2117 \global\let\eq\interline@container\eq\interline@container@clear
2118 \eq\measure@savestate
2119 \eq\display@halign@letcr{#2}%
2120 }

```

**TODO:** describe

```

2121 \def\eq\measure@tag{%
2122 \eq\tagwidth@\z@
2123 \ifdefined\eq\numbering@multi
2124 \if@eqnsw
2125 \eq\tags@container
2126 \eq\tagbox@make\eq\composetag@measure
2127 \ifdefined\eq\tagpos@reserve\else
2128 \eq\tagwidth@\z@
2129 \fi
2130 \fi
2131 \fi
2132 }

```

**TODO:** describe

```

2133 \def\eq\measure@endrow{%
2134 \ifdim\eq\line@prevdepth@=-\@m\p\else
2135 \dimen@\dimexpr\baselineskip-\eq\line@height@-\eq\line@prevdepth@\relax
2136 \ifdim\dimen@<\lineskiplimit
2137 \dimen@\lineskip
2138 \fi
2139 \advance\eq\line@interline@\dimen@
2140 \fi
2141 \eq\dimensions@endrow
2142 \ifdim\eq\tagwidth@>\eq\tagwidth@max@
2143 \global\eq\tagwidth@max@\eq\tagwidth@
2144 \fi
2145 \ifdim\eq\tagwidth@>\z@
2146 \global\advance\eq\tagrows@\@ne
2147 \fi
2148 \global\advance\eq\totalheight@\dimexpr
2149 \eq\line@interline@+\eq\line@height@+\eq\line@depth@
2150 \global\eq\line@interline@\z@
2151 \global\eq\line@prevdepth@\eq\line@depth@
2152 }

```

**TODO:** describe

```

2153 \def\eq\measure@close{%
2154 \advance\eq\row@-\tw@
2155 \eq\totalrows@\eq\row@
2156 \ifnum\eq\totalrows@>\z@
2157 \eq\dimensions@get\@ne
2158 \eq\topheight@\dimexpr\eq\line@height@+\eq\line@interline@\relax
2159 \eq\dimensions@get\eq\totalrows@
2160 \eq\bottomdepth@\eq\line@depth@
2161 \fi
2162 \eq\numbering@measure@blocktag
2163 \begin@group
2164 \eq\tags@container

```

```

2165 \if@eqnsw
2166 \eql@tagbox@make\eql@composetag@measure
2167 \ifdefined\eql@tagpos@reserve\else
2168 \eql@tagwidth@z@
2169 \fi
2170 \eql@dimensions@saveblocktag
2171 \else
2172 \eql@dimensions@savenoblocktag
2173 \eql@numbering@warnunused
2174 \fi
2175 \endgroup
2176 \eql@dimensions@get\z@
2177 \eql@measure@restorestate
2178 }

```

measure@restorestate  
eql@measure@savestate

```

2179 \let\eql@measure@restorestate\@empty
2180 \def\eql@measure@savestate{%
2181 \begingroup
2182 \def\@elt##1{%
2183 \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2184 \global\edef\@gtempa{\cl@ckpt}%
2185 \endgroup
2186 \let\eql@measure@restorestate\@gtempa
2187 }

```

## H.2 Line Breaks

**TODO:** describe

\eql@display@cr

```

2188 \protected\def\eql@display@cr{%
2189 \eql@ampprotecttwo\eql@teststaropt@tight{%
2190 \global\eql@append\eql@interline@container{\eql@displaybreak@pen@MM}%
2191 \eql@display@cr@opt}
2192 \eql@display@cr@opt\z@skip
2193 }

```

\eql@display@cr@opt

```

2194 \def\eql@display@cr@opt[#1]{%
2195 \eql@display@endline
2196 \cr
2197 \noalign{%
2198 \eql@interline@container
2199 \eql@display@injectbefore
2200 \ifnum\eql@displaybreak@pen@=MM
2201 \penalty\interdisplaylinepenalty
2202 \else
2203 \penalty\eql@displaybreak@pen@
2204 \fi
2205 \advance\eql@vspaceskip@glueexpr#1\relax
2206 \vskip\eql@vspaceskip@
2207 \global\advance\eql@line@interline@\eql@vspaceskip@

```

```

2208 \eqldisplay@injectafter
2209 \global\let\eql@interline@container\eql@interline@container@clear
2210 }%
2211 }

```

display@halign@letcr

```

2212 \def\eqldisplay@halign@letcr#1{%
2213 \let\\eqldisplay@cr
2214 \let\eqldisplay@endline#1%
2215 }

```

### H.3 Intertext

**TODO:** describe

**TODO:** revert in everymath?

```

2216 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2217 \eql@amsmath@let\intertext\eql@intertext@default

```

**TODO:** why does it fail in measuring? total width?! determine total width otherwise!

```

2218 \def\eql@intertext@process{%
2219 \eqldisplay@endline
2220 \cr
2221 \ifmeasuring@
2222 \expandafter\@gobble
2223 \else
2224 \expandafter\eql@intertext@print
2225 \fi
2226 }

```

**TODO:** describe **TODO:** prevdepth **TODO:** does this have to be in a vbox? **TODO:** vskip and penalty opposite order **TODO:** can we handle short? certainly needs two passes

```

2227 \def\eql@intertext@print#1{%
2228 \noalign{%
2229 \eqldisplay@halign@end
2230 \let\eql@skip@force@below\z@
2231 \let\eql@skip@force@above\z@
2232 \eql@setkeys{intertext}\eql@intertext@opt
2233 \openup-\eql@spread@
2234 \penalty\postdisplaypenalty
2235 \ifcase\eql@skip@force@below\relax
2236 \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2237 \or
2238 \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2239 \or
2240 \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2241 \or
2242 \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2243 \or
2244 \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2245 \or
2246 \advance\eql@vspaceskip@\z@skip
2247 \or
2248 \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2249 \or
2250 \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax

```

```

2251 \fi
2252 \vskip\eq\vspaceskip@
2253 \global\let\eq\interline@container\eq\interline@container@clear
2254 \vbox{%
2255 \parboxrestore
2256 \ifdim
2257 \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2258 \else
2259 \parshape\@ne
2260 \@totalleftmargin\linewidth
2261 \fi
2262 \noindent
2263 \prevgraf\eq\prevgraf@
2264 \ignorespaces
2265 #1%
2266 \par
2267 \global\eq\prevgraf@\prevgraf
2268 }%
2269 \penalty\predisplaypenalty
2270 \ifcase\eq\skip@force@above\relax
2271 \vskip\glueexpr\eq\skip@long@above\relax
2272 \or
2273 \vskip\glueexpr\eq\skip@short@above\relax
2274 \or
2275 \vskip\glueexpr\eq\skip@cont@above\relax
2276 \or
2277 \vskip\glueexpr\eq\skip@par@above\relax
2278 \or
2279 \vskip\glueexpr\eq\skip@top@above\relax
2280 \or
2281 \vskip\z@skip
2282 \or
2283 \vskip\glueexpr\eq\skip@med@above\relax
2284 \or
2285 \vskip\glueexpr\eq\skip@custom@above\relax
2286 \fi
2287 % \eq\prevdepth@\maxdimen
2288 \eq\prevdepth@\z@
2289 \eq\display@halign@start
2290 }
2291 }

```

**TODO:** describe

```

2292 \newenvironment{eq\intertext}{%
2293 \eq\testopt@tight\eq\intertext@{}}%
2294 }{%
2295 \aftergroup\eq\intertext@after
2296 \ignorespacesafterend
2297 }

```

**TODO:** describe

```

2298 \def\eq\intertext@env{intertext}
2299 \def\eq\intertext@[#1]{%
2300 \global\def\eq\intertext@opt{#1}%
2301 \ifx\@currenvir\eq\intertext@env
2302 \expandafter\eq\scan@env\expandafter\eq\intertext@inject
2303 \else
2304 \expandafter\eq\intertext@process

```

```

2305 \fi
2306 }

```

**TODO:** describe

```

2307 \def\eql@intertext@inject{%
2308   \global\edef\eql@intertext@after{%
2309     \noexpand\eql@intertext@process{%
2310       \ifx\eql@scan@body\eql@scan@body@dump
2311         \eql@scan@body@dump
2312       \else
2313         \noexpand\scantokens{\eql@scan@body@dump}%
2314       \fi
2315     }%
2316   }%
2317 }

```

## H.4 Line Marks

**TODO:** describe

```

2318 \def\eql@markline@pos@below{below}
2319 \def\eql@markline@pos@bottom{bottom}
2320 \def\eql@markline@pos@baseline{baseline}
2321 \let\eql@markline@pos\eql@markline@pos@baseline
2322 \let\eql@markline@shift\z@
2323 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2324 \def\eql@markline@symbol{}

```

**TODO:** describe

```

2325 \def\eql@markline@select#1{%
2326   \let\eql@markline@shift\z@
2327   \eql@setkeys{markline}{#1}%
2328   \eql@markline@print
2329 }

```

**TODO:** describe

```

2330 \def\eql@markline@print{%
2331   \dimen@ \dimexpr\eql@markline@shift\relax
2332   \ifx\eql@markline@pos\eql@markline@pos@below
2333     \ifdim\dimen@=\z@\else
2334       \penalty\@M
2335       \vskip-\dimen@
2336     \fi
2337     \nointerlineskip
2338     \penalty\@M
2339     \vbox{\hfill\hbox{\eql@markline@symbol}}%
2340   \else
2341     \ifx\eql@markline@pos\eql@markline@pos@baseline
2342       \advance\dimen@\prevdepth
2343     \fi
2344     \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}%
2345     \dimen@\prevdepth
2346     \ht\z@\z@
2347     \dp\z@\z@
2348     \nointerlineskip
2349     \penalty\@M

```

```

2350    \vbox{\hfill\box\z}%
2351    \prevdepth\dimen@
2352    \fi
2353 }

```

**TODO:** describe

```

2354 \def\eq@markline@inject#1{%
2355   \let\eq@markline@push\eq@false
2356   \ifx\eq@markline@pos\eq@markline@pos@below\else
2357     \ifdefined\eq@tagsleft\else
2358       \ifx\eq@equations@main\eq@multi@main
2359         \ifdefined\eq@numbering@multi
2360           \if@eqnsw
2361             \let\eq@markline@push\eq@true
2362           \fi
2363         \else
2364           \ifnum\eq@row@=\eq@tagpos@row@
2365             \let\eq@markline@push\eq@true
2366           \fi
2367         \fi
2368       \else
2369         \if@eqnsw
2370           \let\eq@markline@push\eq@true
2371         \fi
2372       \fi
2373     \fi
2374   \fi
2375   \ifdefined\eq@markline@push
2376     \global\eq@append\eq@interline@container{%
2377       \eq@append\eq@display@injectbefore{\eq@markline@select{push,#1}}}%
2378   \else
2379     \global\eq@append\eq@interline@container{%
2380       \eq@append\eq@display@injectbefore{\eq@markline@select{#1}}}%
2381   \fi
2382 }

```

**TODO:** describe

```

2383 \def\eq@markline@amsthm@opt[#1]{\eq@markline@inject{qed,#1}}
2384 \def\eq@markline@amsthm@staropt[#1]{\eq@markline@inject{qed,push,#1}}
2385 \def\eq@markline@amsthm@qed{\eq@teststaropt@tight
2386   \eq@markline@amsthm@staropt\eq@markline@amsthm@opt{}}
2387 \def\eq@markline@amsthm@register#1{\eq@letcs{#1@qed}\eq@markline@amsthm@qed}

```

## I Column Placement

**TODO:** describe

### I.1 Supporting Definitions

$\eq@shape@pos@$  (*dimen*) The registers  $\eq@shape@pos@$  and  $\eq@shape@amount@$  specify the currently selected horizontal alignment (0 for left, 1 for center, 2 for right) and the indentation amount, respectively:

```

2388 \newcount\eq@shape@pos@
2389 \newdimen\eq@shape@amount@

```

```
2390 \let\eq@shape@lastrow\eq@false
```

`\eq@marginleft@` (*dimen*) The registers `\eq@marginleft@` and `\eq@marginright@` store the intended left and right margin for the equation lines: **TODO:** update

```
\eq@marginright@ (dimen)
\eq@centeroffset@ (dimen)
2391 \newdimen\eq@marginleft@
2392 \newdimen\eq@marginright@
2393 \newdimen\eq@marginleft@min@
2394 \newdimen\eq@centeroffset@
```

## I.2 Shape Schemes

The horizontal alignment of each line is specified by a shape scheme.

`\eq@shape@tab@...` We select the scheme through a `\csname` selector with the following names:

```
2395 \def\eq@shape@tab@default{default}
2396 \def\eq@shape@tab@left{left}
2397 \def\eq@shape@tab@center{center}
2398 \def\eq@shape@tab@right{right}
2399 \def\eq@shape@tab@first{first}
2400 \def\eq@shape@tab@hanging{hanging}
2401 \def\eq@shape@tab@steps{steps}
```

For convenience, we add further alias names for the schemes:

```
2402 \let\eq@shape@tab@def\eq@shape@tab@default
2403 \let\eq@shape@tab@l\eq@shape@tab@left
2404 \let\eq@shape@tab@c\eq@shape@tab@center
2405 \let\eq@shape@tab@r\eq@shape@tab@right
2406 \let\eq@shape@tab@f\eq@shape@tab@first
2407 \let\eq@shape@tab@i\eq@shape@tab@first
2408 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2409 \let\eq@shape@tab@lc\eq@shape@tab@hanging
2410 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2411 \let\eq@shape@tab@lcr\eq@shape@tab@steps
2412 \let\eq@shape@tab@lcr\eq@shape@tab@steps
```

`\eq@shape@mode` The currently selected scheme is stored in `\eq@shape@mode`. It is set to default:

```
2413 \let\eq@shape@mode\eq@shape@tab@default
```

`\eq@shape@set` Set the scheme via the translation table:

```
2414 \def\eq@shape@set#1{%
2415   \ifcsname eq@shape@tab@#1\endcsname
2416     \expandafter\let\expandafter\eq@shape@mode
2417       \csname eq@shape@tab@#1\endcsname
2418   \else
2419     \eq@error{shape '#1' unknown: setting to default}%
2420     \let\eq@shape@mode\eq@shape@tab@default
2421   \fi
2422 }
```

`\eq@shape@layoutcenter@...` Define the uniform shape schemes `left`, `center`, `right` and `default` for the central and `\eq@shape@layoutleft@...` left alignment layout. The scheme functions determine the desired alignment and indentation for the current row:



```

2423 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
2424 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne\eq@shape@amount@z@}
2425 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2426 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2427 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
2428 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne\eq@shape@amount@z@}
2429 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2430 \let\eq@shape@layoutleft@default\eq@shape@layoutleft@left

```

The **first** scheme implements left alignment with indentation for the first line (unless there is only one line):

```

2431 \def\eq@shape@layoutcenter@first{%
2432   \eq@shape@pos@z@
2433   \eq@shape@amount@z@
2434   \ifnum\eq@totalrows@>\@ne
2435     \ifnum\eq@row@=\@ne
2436       \eq@shape@amount@\eq@indent@
2437     \fi
2438   \fi
2439 }
2440 \def\eq@shape@layoutleft@first{%
2441   \eq@shape@pos@z@
2442   \eq@shape@amount@z@
2443   \ifnum\eq@totalrows@>\@ne
2444     \ifnum\eq@row@=\@ne
2445       \eq@shape@amount@\eq@indent@
2446     \fi
2447   \fi
2448 }

```

The **hanging** scheme implements left alignment with hanging indentation for the first line (unless there is only one line). In central alignment layout all but the first line are indented while in left aligned layout the first line has negative indentation:

```

2449 \def\eq@shape@layoutcenter@hanging{%
2450   \eq@shape@pos@z@
2451   \eq@shape@amount@\eq@indent@
2452   \ifnum\eq@totalrows@>\@ne
2453     \ifnum\eq@row@=\@ne
2454       \eq@shape@amount@z@
2455     \fi
2456   \fi
2457 }
2458 \def\eq@shape@layoutleft@hanging{%
2459   \eq@shape@pos@z@
2460   \eq@shape@amount@z@
2461   \ifnum\eq@totalrows@>\@ne
2462     \ifnum\eq@row@=\@ne
2463       \eq@shape@amount@-\eq@indent@
2464     \fi
2465   \fi
2466 }

```

The **steps** scheme implements singles out the first and last lines which are shifted left and right, respectively. In central alignment layout the shift operates on the alignment whereas in left alignment layout the shift uses indentation:

```

2467 \def\eq@shape@layoutcenter@steps{%
2468   \eq@shape@amount@z@

```

```

2469 \eq@shape@pos@ \@ne
2470 \ifnum\eq@totalrows@>\@ne
2471   \ifnum\eq@row@=\@ne
2472     \eq@shape@pos@ \z@
2473   \fi
2474   \ifnum\eq@row@=\eq@totalrows@
2475     \eq@shape@pos@ \tw@
2476   \fi
2477 \fi
2478 }
2479 \def\eq@shape@layoutleft@steps{%
2480   \eq@shape@pos@ \z@
2481   \eq@shape@amount@ \z@
2482   \ifnum\eq@totalrows@>\@ne
2483     \ifnum\eq@row@=\@ne
2484       \eq@shape@amount@-\eq@indent@
2485     \fi
2486     \ifnum\eq@row@=\eq@totalrows@
2487       \eq@shape@amount@\eq@indent@
2488     \fi
2489   \fi
2490 }

```

`\eq@shape@select` Select the shape selector function for the current scheme `@\eq@shape@mode` and layout  
`\eq@shape@eval` and store it in `\eq@shape@eval`:

```

2491 \let\eq@shape@eval\@undefined
2492 \def\eq@shape@select{%
2493   \expandafter\let\expandafter\eq@shape@eval
2494   \csname eq@shape%
2495     @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2496     @\eq@shape@mode\endcsname
2497 }

```

`\eq@shape@alignleft` Adjust the alignment of the current equation line. The optional argument specifies the  
`\eq@shape@alignright` amount of indentation:  
`\eq@shape@aligncenter`

```

2498 \protected\def\eq@shape@alignleft{%
2499   \global\eq@append\eq@cell@container{\eq@shape@pos@ \z@}%
2500   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2501 \protected\def\eq@shape@aligncenter{%
2502   \global\eq@append\eq@cell@container{\eq@shape@pos@ \@ne}%
2503   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2504 \protected\def\eq@shape@alignright{%
2505   \global\eq@append\eq@cell@container{\eq@shape@pos@ \tw@}%
2506   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2507 \def\eq@shape@align@testpar#1{%
2508   \eq@ifstar@tight{#1[\eq@indent@]}%
2509   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2510   {\eq@testopt@tight{#1}\z@}}
2511 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

`\eq@shape@alignamount` **TODO:** describe

```

2512 \protected\def\eq@shape@alignamount{%
2513   \eq@ampprotecttwo\eq@ifstar@tight
2514   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2515 \def\eq@shape@alignamount@add#1{%
2516   \global\eq@appendexpand\eq@cell@container{

```

```

2517 \advance\eq\shape@amount@\the\glueexpr#1\relax\relax}}
2518 \def\eq\shape@alignamount@set#1{%
2519 \global\eq\appendexpand\eq\cell@container{%
2520 \eq\shape@amount@\the\glueexpr#1\relax\relax}}
2521 \def\eq\shape@align@enable{%
2522 \let\shoveleft\eq\shape@alignleft
2523 \let\shovecenter\eq\shape@aligncenter
2524 \let\shoveright\eq\shape@alignright
2525 \let\shoveby\eq\shape@alignamount
2526 }

```

**TODO:** describe

```

2527 \protected\def\eq\shape@align@default{%
2528 \eq\warn@here{\shove...}%
2529 \eq\ampprotect\eq\shape@align@testpar\eq@gobbleopt}
2530 \protected\def\eq\shape@alignamount@default{%
2531 \eq\warn@here{\shove...}%
2532 \eq\ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2533 \def\eq\shape@align@disable{%
2534 \let\shoveleft\eq\shape@align@default
2535 \let\shovecenter\eq\shape@align@default
2536 \let\shoveright\eq\shape@align@default
2537 \let\shoveby\eq\shape@alignamount@default
2538 }

```

## I.3 Width Data

`\width@block@` (*dimen*)

```

2539 \newdimen\eq\tagwidth@block@
2540 \newdimen\eq\tagheight@block@
2541 \newdimen\eq\tagdepth@block@

```

`\eq\dimensions@tab` **TODO:** new

```

2542 \let\eq\dimensions@tab\@empty

```

`\eq\dimensions@reset`

```

2543 \def\eq\dimensions@reset{%
2544 \let\eq\dimensions@tab\@empty
2545 \eq\tagwidth@max@\z@
2546 \eq\tagrows@\z@
2547 }

```

`\eq\dimensions@add`

```

2548 \def\eq\dimensions@add#1{%
2549 \global\eq\appendexpand\eq\dimensions@tab{#1}%
2550 }

```

`\eq\dimensions@addreg`

```

2551 \def\eq\dimensions@addreg#1{#1\the#1\relax}

```

`\eq\dimensions@startrow`

```

2552 \def\eq\dimensions@startrow{%

```

```

2553 \eqldimensions@add{\eqldimensions@addreg\eqldrow}%
2554 }

```

@dimensions@savecell

```

2555 \def\eqldimensions@savecell{%
2556 \eqldimensions@add{%
2557 \eqldimensions@addreg\eqldshape@pos@
2558 \eqldimensions@addreg\eqldcellwidth@
2559 \eqldimensions@addreg\eqldshape@amount@
2560 \noexpand\eqldimensions@cellcall
2561 }%
2562 }

```

l@dimensions@savesep

```

2563 \def\eqldimensions@savesep{%
2564 \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2565 }

```

ql@dimensions@endrow

```

2566 \def\eqldimensions@endrow{%
2567 \eqldimensions@add{,%
2568 \eqldimensions@addreg\eqldtagwidth@
2569 \eqldimensions@addreg\eqldline@height@
2570 \eqldimensions@addreg\eqldline@depth@
2571 \eqldimensions@addreg\eqldline@interline@
2572 ;}%
2573 }

```

ensions@saveblocktag

```

2574 \def\eqldimensions@saveblocktag{%
2575 \eqldimensions@add{\eqldrow@0\relax,%
2576 \eqldtagwidth@block@the\eqldtagwidth@\relax
2577 \eqldtagheight@block@the\ht\eqldtagbox@\relax
2578 \eqldtagdepth@block@the\dp\eqldtagbox@\relax
2579 \eqldimensions@addreg\eqldtagpos@shift@
2580 \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
2581 \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
2582 ;}%
2583 \global\eqldtagwidth@max@eqldtagwidth@
2584 \global\eqldtaggrows@\@ne
2585 }

```

sions@savenoblocktag

```

2586 \def\eqldimensions@savenoblocktag{%
2587 \eqldimensions@add{\eqldrow@0\relax,;%
2588 }

```

\eqldimensions@for

```

2589 \def\eqldimensions@for#1{%
2590 \def\eqldimensions@forcall{#1}%
2591 \expandafter\eqldimensions@forstep\eqldimensions@tab
2592 }

```

l@dimensions@forstep

```

2593 \def\eql@dimensions@forstep\eql@row@#1\relax#2,##3;%
2594 \eql@row@#1\relax
2595 \ifnum\eql@row@=\z@\else
2596   #3%
2597   \def\eql@dimensions@cells{##2}%
2598   \eql@dimensions@forall
2599   \expandafter\eql@dimensions@forstep
2600 \fi
2601 }

```

\eql@dimensions@get

```

2602 \def\eql@dimensions@get#1{%
2603 \eql@row@#1\relax
2604 \expandafter\eql@dimensions@getdef\expandafter{\the\eql@row@}%
2605 \expandafter\eql@dimensions@getparse\eql@dimensions@tab\@nil
2606 }

```

ql@dimensions@getdef

```

2607 \def\eql@dimensions@getdef#1{%
2608 \def\eql@dimensions@getparse
2609   ##1\eql@row@#1\relax##2,##3;##4\@nil{%
2610   ##3%
2611   \def\eql@dimensions@cells{##2}%
2612   }%
2613 }

```

\eql@colwidth@tab

```

2614 \let\eql@colwidth@tab\@empty

```

\eql@colwidth@get

```

2615 \def\eql@colwidth@get#1{%
2616 \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2617 }

```

\eql@colwidth@save

```

2618 \def\eql@colwidth@save#1{%
2619 \edef\eql@colwidth@tab{%
2620 \noexpand\or\the#1%
2621 \unexpanded\expandafter{\eql@colwidth@tab}%
2622 }%
2623 }

```

\eql@dimensions@calc Compute the space that is available at the beginning and at the end of the row stored in \eql@dimensions@cells. The space available at the beginning is returned in \eql@line@avail@. and \eql@line@availsep@ describes the number of unused intercolumn separations. The total used width is returned in \eql@line@width@ and \eql@line@widthsep@ describes the number of used intercolumn separations. The available space at the end of the row is given as the difference to \eql@totalwidth@:

```

2624 \def\eql@dimensions@calc{%
2625 \eql@column@\z@
2626 \eql@line@pos@\z@

```

```

2627 \eq@line@possep@ \z@
2628 \eq@line@avail@ \eq@totalwidth@
2629 \eq@line@availsep@ \eq@intercolumns@
2630 \eq@line@width@ \z@
2631 \eq@line@widthsep@ \z@
2632 \let\eq@dimensions@cellcall\eq@dimensions@calc@call
2633 \let\eq@dimensions@sepcall\eq@dimensions@calc@callsep
2634 \eq@dimensions@cells
2635 }

```

`ensions@calc@callsep` Callback for each intercolumn space.

```

2636 \def\eq@dimensions@calc@callsep{%
2637   \advance\eq@line@possep@ \@ne
2638 }%

```

`dimensions@calc@call` Callback for each column. When a non-blank cell is encountered, the available space on the left will be fixed if it is still undetermined, and the total width is updated to the current position: **TODO:** implement an offset for central alignment (global?!)

```

2639 \def\eq@dimensions@calc@call{%
2640   \advance\eq@column@ \@ne
2641   \ifnum\eq@totalcolumns@=\@ne
2642     \dimen@ \eq@totalwidth@
2643   \else
2644     \dimen@ \eq@colwidth@get\eq@column@ \relax
2645   \fi
2646   \ifdim\eq@cellwidth@>\z@
2647     \ifdim\eq@line@width@=\z@
2648       \eq@line@avail@ \eq@line@pos@
2649       \eq@line@availsep@ \eq@line@possep@
2650       \ifcase\eq@shape@pos@
2651       \or
2652         \advance\eq@line@avail@ \dimexpr
2653           (\dimen@-\eq@cellwidth@+\eq@centeroffset@)/\tw@ \relax
2654       \or
2655         \advance\eq@line@avail@ \dimexpr\dimen@-\eq@cellwidth@ \relax
2656       \fi
2657       \advance\eq@line@avail@ \eq@shape@amount@
2658     \fi
2659     \eq@line@width@ \eq@line@pos@
2660     \eq@line@widthsep@ \eq@line@possep@
2661     \ifcase\eq@shape@pos@
2662     \advance\eq@line@width@ \eq@cellwidth@
2663     \or
2664     \advance\eq@line@width@ \dimexpr
2665       (\dimen@+\eq@cellwidth@+\eq@centeroffset@)/\tw@ \relax
2666     \or
2667     \advance\eq@line@width@ \dimen@
2668     \fi
2669     \advance\eq@line@width@ \eq@shape@amount@
2670   \fi
2671   \advance\eq@line@pos@ \dimen@
2672 }

```

## I.4 Best Line Selection

`@numbering@best@auto` **TODO:** describe

```
2673 \let\eql@numbering@best@auto\eql@false
```

`@best@row@` (*counter*)

`@best@space@` (*dimen*)

`@numbering@best@use` (*bool*)

```
2674 \newcount\eql@numbering@best@row@
```

```
2675 \newdimen\eql@numbering@best@space@
```

```
2676 \let\eql@numbering@best@use\eql@false
```

`@numbering@best@find` Determine the row with the largest available space on the side of the tags:

```
2677 \def\eql@numbering@best@find{%
2678   \eql@numbering@best@row@ \z@
2679   \eql@numbering@best@space@ \z@
2680   \eql@dimensions@for{%
2681     \eql@dimensions@calc
2682     \ifdefined\eql@tagsleft
2683       \dimen@ \eql@line@avail@
2684     \else
2685       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
2686     \fi
2687     \ifdim \dimen@ > \eql@numbering@best@space@
2688       \eql@numbering@best@row@ \eql@row@
2689       \eql@numbering@best@space@ \dimen@
2690     \fi
2691   }%
2692   \ifnum \eql@numbering@best@row@ > \z@
2693     \eql@tagpos@row@ \eql@numbering@best@row@
2694     \let\eql@tagpos@continuous\eql@false
2695     \eql@tagpos@prevrow@ \z@
2696   \fi
2697 }
```

`@numbering@best@test` **TODO:** describe

```
2698 \def\eql@numbering@best@test#1{%
2699   \eql@dimensions@get#1%
2700   \eql@dimensions@calc
2701   \ifdefined\eql@tagsleft
2702     \dimen@ \dimexpr \eql@line@avail@
2703       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
2704   \else
2705     \dimen@ \dimexpr \displaywidth@ - \eql@line@width@
2706       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
2707   \fi
2708   \ifdim \dimen@ < \eql@tagwidth@block@
2709     \let\eql@numbering@best@use\eql@true
2710   \fi
2711 }
```

`@numbering@best@eval` **TODO:** describe **TODO:** to test both lines individually may cause undesired effects

```
2712 \def\eql@numbering@best@eval{%
2713   \ifdefined\eql@numbering@best@auto
2714     \ifdefined\eql@numbering@best@use\else
2715       \ifdefined\eql@numbering@multi\else
```

```

2716         \ifnum\eq\tagpos@row@>\z@
2717         \eq\numbering@best@test\eq\tagpos@row@
2718         \fi
2719         \ifnum\eq\tagpos@prevrow@>\z@
2720         \eq\numbering@best@test\eq\tagpos@prevrow@
2721         \fi
2722     \fi
2723 \fi
2724 \fi
2725 \ifdefined\eq\numbering@best@use
2726     \eq\numbering@best@find
2727 \fi
2728 }

```

## I.5 Tag Margin

**TODO:** describe **TODO:** if a tag margin is installed for a single line, it will shift the center even if there is no tag or importantly if a tag has been raised.

djust@calc@tagmargin

```

2729 \def\eq\adjust@calc@tagmargin{%
2730     \ifdefined\eq\tagmargin@val
2731         \eq\tagmargin@\glueexpr\eq\tagmargin@val\relax
2732     \else
2733         \eq\tagmargin@\eq\tagwidth@max@
2734         \ifdim\eq\tagmargin@>\z@
2735             \advance\eq\tagmargin@-\eq\tagsepmin@
2736         \fi
2737     \fi

2738     \dimen@\eq\tagrows@\p@
2739     \ifnum\eq\totalrows@=\@ne
2740         \ifnum\eq\tagrows@=\@ne
2741             \advance\dimen@1sp\relax
2742         \fi
2743     \fi
2744     \ifdim\dimen@>\eq\totalrows@\eq\tagmargin@ratio@\else
2745         \eq\tagmargin@\z@
2746     \fi

2747     \@tempdima\dimexpr\displaywidth
2748         -\eq\totalwidth@-\eq\intercolumns@\eq\colsepmin@\relax
2749     \@tempdimb\dimexpr\@tempdima-\tw@\eq\tagmargin@\relax
2750     \ifdim\@tempdimb>\z@
2751         \ifdim\eq\tagmargin@threshold\@tempdima<\@tempdimb
2752             \eq\tagmargin@\z@
2753         \fi
2754     \fi
2755 }

```

## I.6 Single Column

ql@adjust@calc@lines

```

2756 \def\eq\adjust@calc@lines{%
2757     \eq\totalcolumns@\@ne

```



```

2758 \eql@intercolumns@\z@
2759 \eql@colsep@\z@
2760 \ifdefined\eql@layoutleft
2761 \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2762 \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2763 \ifdim\eql@marginleft@<\eql@marginleft@min@
2764 \eql@marginleft@\eql@marginleft@min@
2765 \fi
2766 \dimen@\glueexpr\eql@layoutleftmarginmax\relax
2767 \ifdim\eql@marginleft@>\dimen@
2768 \eql@marginleft@\dimen@
2769 \fi
2770 \eql@marginright@\z@
2771 \eql@centeroffset@\z@
2772 \else
2773 \eql@adjust@calc@tagmargin
2774 \ifdefined\eql@paddingleft@val
2775 \eql@marginleft@\dimexpr
2776 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2777 -\glueexpr\eql@paddingleft@val\relax\relax
2778 \ifdim\eql@marginleft@<\z@
2779 \eql@marginleft@\z@
2780 \fi
2781 \else
2782 \eql@marginleft@\z@
2783 \fi
2784 \ifdefined\eql@paddingright@val
2785 \eql@marginright@\dimexpr
2786 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2787 -\glueexpr\eql@paddingright@val\relax\relax
2788 \ifdim\eql@marginright@<\z@
2789 \eql@marginright@\z@
2790 \fi
2791 \else
2792 \eql@marginright@\z@
2793 \fi
2794 \ifdim\eql@tagmargin@>\z@
2795 \ifdefined\eql@tagsleft
2796 \ifdim\eql@marginleft@<\eql@tagsepmin@
2797 \eql@marginleft@\eql@tagsepmin@
2798 \fi
2799 \advance\eql@marginleft@\eql@tagmargin@
2800 \advance\eql@centeroffset@\eql@tagmargin@
2801 \else
2802 \ifdim\eql@marginright@<\eql@tagsepmin@
2803 \eql@marginright@\eql@tagsepmin@
2804 \fi
2805 \advance\eql@marginright@\eql@tagmargin@
2806 \advance\eql@centeroffset@-\eql@tagmargin@
2807 \fi
2808 \fi
2809 \eql@marginleft@min@\z@
2810 \eql@centeroffset@\dimexpr\eql@marginright@-\eql@marginleft@
2811 \ifdefined\eql@tagsleft+ \else -\fi \eql@tagmargin@ \relax
2812 \fi

2813 \eql@totalwidth@\dimexpr\displaywidth
2814 -\eql@marginleft@-\eql@marginright@ \relax
2815 }

```

## I.7 Multiple Columns

The following code computes the horizontal placement of columns. It distributes the columns evenly according to the layout presets and then determines whether there is enough space to place an equation tag on each line. If not, the intercolumn spacing and the space at the opposite margin can be reduced.

`@adjust@calc@columns` Main method to adjust column placement and spacing:

```
2816 \def\eql@adjust@calc@columns{%
```

If there is just a single alignment structure, there will be no intercolumn space that might stretch to adjust the columns to the margins. We disable fulllength to avoid a division by zero. Also guard against no columns at all (empty body), just in case:

```
2817 \ifnum\eql@totalcolumns@<\thr@@
2818   \eql@totalcolumns@\tw@
2819   \let\eql@columns@fulllength\eql@false
2820 \fi
```

Determine the number of intercolumn spaces `\eql@intercolumns@`:

```
2821 \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2822 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

Determine the left or target margin width depending on the layout:

```
2823 \ifdefined\eql@layoutleft
2824   \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2825   \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2826   \ifdim\eql@marginleft@<\eql@marginleft@min@
2827     \eql@marginleft@\eql@marginleft@min@
2828   \fi
2829 \else
```

Get the desired tag margin, increase by minimum tag separation if columns are aligned to the margins. Cancel tag margin if too wide:

```
2830   \eql@adjust@calc@tagmargin
2831   \ifdefined\eql@columns@fulllength
2832     \ifdim\eql@tagmargin@>\z@
2833       \advance\eql@tagmargin@\eql@tagsepmin@
2834     \fi
2835   \fi
2836   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2837     -\eql@intercolumns@\eql@colsepmin@\relax
2838     \eql@tagmargin@\z@
2839   \fi
2840   \eql@marginleft@min@\z@
2841 \fi
```

Compute the intercolumn space `\eql@colsep@`:

```
2842 \ifnum\eql@intercolumns@>\z@
```

Distribute the available horizontal space evenly onto the intercolumn spaces and the margins. Unless the columns are aligned to the margins, there are two margins in central alignment layout but only the right margin in left alignment layout:

```

2843 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2844 \ifdefined\eql@layoutleft
2845 \advance\eql@colsep@-\eql@marginleft@
2846 \else
2847 \advance\eql@colsep@-\eql@tagmargin@
2848 \fi
2849 \count@\eql@intercolumns@
2850 \ifdefined\eql@columns@fulllength\else
2851 \ifdefined\eql@layoutleft
2852 \advance\count@\@ne
2853 \else
2854 \advance\count@\tw@
2855 \fi
2856 \fi
2857 \divide\eql@colsep@\count@

```

Ensure that the intercolumn separation is within the specified bounds. Disable the upper bound if columns are to be aligned to the margins:

```

2858 \ifdim\eql@colsep@<\eql@colsepmin@
2859 \eql@colsep@\eql@colsepmin@
2860 \else
2861 \ifdefined\eql@columns@fulllength\else
2862 \dimen@\glueexpr\eql@colsepmax@val\relax
2863 \ifdim\eql@colsep@>\dimen@
2864 \eql@colsep@\dimen@
2865 \fi
2866 \fi
2867 \fi
2868 \else

```

For a single column, set the column separation to the minimum amount:

```

2869 \eql@colsep@\eql@colsepmin@
2870 \fi

```

Compute the left margin `\eql@marginleft@` depending on the layout:

```

2871 \ifdefined\eql@layoutleft

```

Set the default value:

```

2872 \ifdim\eql@colsep@=\eql@colsepmin@

```

If in left alignment layout the intercolumn space has been adjusted, compute the available space, determine left margin and make sure it is between the minimum and the default value:

```

2873 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
2874 -\eql@intercolumns@\eql@colsep@\relax
2875 \ifdim\dimen@<\eql@marginleft@
2876 \ifdim\dimen@<\eql@marginleft@min@
2877 \eql@marginleft@\eql@marginleft@min@
2878 \else
2879 \eql@marginleft@\dimen@
2880 \fi
2881 \fi
2882 \fi
2883 \else

```

In central alignment mode with column aligned to the margins, set margin to zero:

```

2884 \ifdefined\eql@columns@fulllength
2885 \eql@marginleft@<z@

```

In central alignment mode with margins, distribute the available space equally to both margins, or remove the left margin if insufficient:

```

2886 \else
2887 \eql@marginleft@<\dimexpr(\displaywidth-\eql@totalwidth@
2888 -\eql@intercolumns@<\eql@colsep@-\eql@tagmargin@)/\tw@<\relax
2889 \ifdim\eql@marginleft@<<z@
2890 \eql@marginleft@<z@
2891 \fi
2892 \fi

```

Add tag margin in case of left tags:

```

2893 \ifdefined\eql@tagsleft
2894 \advance\eql@marginleft@<\eql@tagmargin@
2895 \fi
2896 \fi

```

Find the best row for tag placement:

```

2897 \eql@numbering@best@eval

```

Next consider all rows with tags and adjust the intercolumn and margin space to make the tags fit into the available space at the corresponding side as far as possible. First, select code depending on tag placement:

```

2898 \ifdefined\eql@tagsleft
2899 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
2900 \else
2901 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2902 \fi

```

Loop over all rows or select the single row containing the tag. Fetch the width data for the current row. If a tag is present, compute the available space and try to adjust spaces if needed: **TODO:** complete for prevrow, ideally join treatment

```

2903 \ifdefined\eql@numbering@multi
2904 \eql@dimensions@for{%
2905 \ifdim\eql@tagwidth@>z@
2906 \eql@dimensions@calc
2907 \eql@adjust@columns@test
2908 \fi
2909 }%
2910 \else
2911 \ifnum\eql@tagpos@row@>z@
2912 \ifnum\eql@tagpos@row@>\eql@totalrows@\else
2913 \eql@dimensions@get\eql@tagpos@row@
2914 \eql@tagwidth@\eql@tagwidth@block@
2915 \eql@dimensions@calc
2916 \eql@adjust@columns@test
2917 \fi
2918 \fi
2919 \ifnum\eql@tagpos@prevrow@>z@
2920 \eql@dimensions@get\eql@tagpos@prevrow@
2921 \eql@tagwidth@\eql@tagwidth@block@
2922 \eql@dimensions@calc
2923 \eql@adjust@columns@test
2924 \fi

```

```
2925 \fi
```

From now on `\eq@totalwidth@` will include the left margin and the total intercolumn separation:

```
2926 \advance\eq@totalwidth@\dimexpr
2927 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
2928 }
```

### Placement for Right Tags.

`\columns@test@tagsright` Test whether the spacing can be adjusted to make the current row fit:

```
2929 \def\eq@adjust@columns@test@tagsright{%
```

The register `\@tempdima` will hold the amount of available space. **TODO:** does this apply equally to left alignment layout?

```
2930 \@tempdima\dimexpr\displaywidth-\eq@linewidth-\eq@tagwidth@\relax
```

Test whether the space at the end of the row is sufficient to hold the tag with the current settings.

```
2931 \ifdim\@tempdima<\dimexpr
2932 \eq@marginleft@+\eq@linewidthsep@\eq@colsep@\relax
```

If not, determine whether the row and tag may at all fit into the available space with minimal intercolumn spaces and minimal left margin (in left alignment layout).

```
2933 \ifdim\@tempdima<\dimexpr
2934 \eq@marginleft@min@+\eq@linewidthsep@\eq@colsepmin@\relax\else
```

If so, hand over to `\eq@adjust@columns@modify@tagsright`.

```
2935 \eq@adjust@columns@modify@tagsright
2936 \fi
2937 \fi
2938 }
```

`\columns@modify@tagsright` Adjust the intercolumn space and left margin to make the row fit.

```
2939 \def\eq@adjust@columns@modify@tagsright{%
```

If there are any intercolumn spaces that contribute to the available space, determine how much intercolumn separation would be needed while keeping the current left margin fixed (in left alignment layout). In central alignment layout, assume that the left margin will be adjusted to match the intercolumn separation by stepping the number of columns to divide by.

```
2940 \ifnum\eq@linewidthsep@>\z@
2941 \dimen@\@tempdima
2942 \count@\eq@linewidthsep@
2943 \ifdefined\eq@layoutleft
2944 \advance\dimen@-\eq@marginleft@
2945 \else
2946 \ifdefined\eq@columns@fulllength\else
2947 \advance\count@\@ne
2948 \fi
2949 \fi
2950 \divide\dimen@\count@
```

If smaller, reduce the intercolumn separation, but make sure to not exceed the minimum allowed value.

```

2951 \ifdim\dimen@<\eq\colsep@
2952 \ifdim\dimen@<\eq\colsepmin@
2953 \eq\colsep@\eq\colsepmin@
2954 \else
2955 \eq\colsep@\dimen@
2956 \fi
2957 \fi
2958 \fi

```

Now adjust the left margin as much as needed to fit the contents.

```

2959 \dimen@\dimexpr\@tempdima-\eq\line@widthsep@\eq\colsep@\relax
2960 \ifdim\eq\marginleft@>\dimen@
2961 \eq\marginleft@\dimen@
2962 \fi
2963 }

```

### Placement for Left Tags.

`columns@test@tagsleft` Test whether the spacing can be adjusted to make the current row fit:

```

2964 \def\eq\adjust@columns@test@tagsleft{%

```

The register `\@tempdima` will hold the deficit amount of space at the beginning of the row without adjustable space, and the register `\count@` will hold the number of intercolumn spaces that would contribute to space adjustments.

```

2965 \count@\numexpr\eq\intercolumns@-\eq\line@availsep@\relax
2966 \@tempdima\dimexpr\eq>tagwidth@-\eq\line@avail@\relax

```

Test whether the space at the beginning of the row is sufficient to hold the tag with the current settings.

```

2967 \ifdim\@tempdima>\dimexpr
2968 \eq\marginleft@+\eq\line@availsep@\eq\colsep@\relax

```

If not, first verify that the tag will fit the line (or the maximal left margin in left alignment layout).

```

2969 \ifdim\eq>tagwidth@<%
2970 \ifdefined\eq\layoutleft
2971 \glueexpr\eq\layoutleftmarginmax\relax
2972 \else
2973 \displaywidth
2974 \fi

```

If so, determine whether the row and tag may at all fit into the available space with minimal intercolumn spaces.

```

2975 \ifdim\@tempdima>\dimexpr
2976 \displaywidth-\eq\totalwidth@-\count@\eq\colsepmin@\relax\else

```

If so, hand over to `\eq\adjust@columns@modify@tagsleft`.

```

2977 \eq\adjust@columns@modify@tagsleft
2978 \fi
2979 \fi
2980 \fi
2981 }

```

umns@modify@tagsleft Adjust the intercolumn space and left margin to make the row fit.

```
2982 \def\eql@adjust@columns@modify@tagsleft{%
```

If there are any intercolumn spaces that contribute to the available space, determine how much intercolumn separation would be needed while keeping the current right margin fixed. In central alignment layout, assume that the right margin will be adjusted to match the intercolumn separation by stepping the number of columns to divide by.

```
2983 \ifnum\count@>\z@
2984 \dimen@ \dimexpr\displaywidth-\eql@totalwidth@-\@tempdima\relax
2985 \ifdefined\eql@columns@fulllength\else
2986 \advance\count@\@ne
2987 \fi
2988 \divide\dimen@\count@
```

If smaller, reduce the intercolumn separation, but make sure to not exceed the minimum allowed value. Also adjust the left margin to keep the right margin fixed.

```
2989 \ifdim\dimen@<\eql@colsep@
2990 \ifdim\dimen@<\eql@colsepmin@
2991 \dimen@\eql@colsepmin@
2992 \fi
2993 \advance\dimen@-\eql@colsep@
2994 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
2995 \advance\eql@colsep@\dimen@
2996 \fi
2997 \fi
```

Now adjust the left margin as much as needed to fit the contents.

```
2998 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
2999 \ifdim\eql@marginleft@<\dimen@
3000 \eql@marginleft@\dimen@
3001 \fi
3002 }
```

## J Single Column Arrangement

The following code adjusts individual lines of equations for the equation and lines mode according to the selected layout and shape.

### J.1 Supporting Definitions

\inf@bad The \inf@bad constant is for testing overfull boxes:

```
3003 \ifdefined\inf@bad\else%
3004 \newcount\inf@bad
3005 \inf@bad1000000\relax
3006 \fi
```

\eql@restore@hfuzz We need to change the value of \hfuzz temporarily. The method \eql@save@hfuzz stores the value for recovery through \eql@restore@hfuzz:

```
3007 \let\eql@restore@hfuzz\@empty
3008 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

`\eql@alignbadness@` The registers `\eql@alignbadness@` and `\eql@tagbadness@` store the allowable badness threshold for shrinking equation lines to the intended margin or to fit into the line at all before the tag is raised or lowered:

```
3009 \newcount\eql@alignbadness@
3010 \newcount\eql@tagbadness@
3011 \newcount\eql@arrange@badness@
3012 \eql@alignbadness@ \inf@bad
3013 \eql@tagbadness@ \inf@bad
```

## J.2 Arrangement Methods

`\eql@arrange@try` Try to fit the current equation line in the available space. Argument #1 specifies the amount of reserved space. Unpack the box `\eql@cellbox@`, replace the previous kerning with the new reserved space, and save the box back into `\eql@cellbox@`:

```
3014 \def\eql@arrange@try#1{%
3015   \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax
3016     \setbox\eql@cellbox@\hbox to\displaywidth{%
3017       \unhbox\eql@cellbox@\unkern\kern#1}%
3018     \eql@arrange@badness@\badness
3019   \else
3020     \eql@arrange@badness@\m@ne
3021   \fi
3022 }
```

`\eql@arrange@print` We have found the final adjustment of the current line, so we typeset it with initial and final space adjustments #1 and #2, respectively. Restore the original value for `\hfuzz`:  
**TODO:** adjust

```
3023 \def\eql@arrange@print#1#2{%
3024   \eql@restore@hfuzz
3025   \if@eqnsw
3026     \ifdefined\eql@tagsleft
3027       \eql@tagbox@print@tagsleft
3028     \fi
3029   \fi
3030   \hbox to\displaywidth{%
3031     #1%
3032     \unhbox\eql@cellbox@\unkern
3033     #2%
3034     \eql@tagging@mathaddlast
3035   }%
3036   \if@eqnsw
3037     \ifdefined\eql@tagsleft\else
3038       \eql@tagbox@print@tagsright
3039     \fi
3040   \fi
3041 }
```

`\eql@arrange@print@alignleft` Fit the current equation line with the selected alignment within a given left and right margins #1 and #2. If we're on the first line, adjust `\eql@display@firstavail@` to the minimum left available space we can guarantee:

```
3042 \def\eql@arrange@print@alignleft#1#2{%
3043   \eql@display@firstavail@set{\dimexpr#1\relax}%
3044   \eql@arrange@print{\kern#1}{\kern#2}%
3045 }
```



```

3046 \def\eql@arrange@print@alignright#1#2{%
3047   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3048   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3049 }

```

```

3050 \def\eql@arrange@print@aligncenter#1{%
3051   \eql@display@firstavail@set{\dimexpr
3052     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3053   \ifdim#1>\z@
3054     \eql@arrange@print{\kern#1\hfil}{}%
3055   \else
3056     \eql@arrange@print{\hfil}{\kern-#1}%
3057   \fi
3058 }

```

`\eql@arrange@init` Initialise the horizontal adjustment framework. Turn off overfull box messages temporarily – otherwise there would be unwanted extra ones emitted during our measuring operations. Select the shape scheme:

```

3059 \def\eql@arrange@init{%
3060   \eql@save@hfuzz
3061   \hfuzz\maxdimen
3062   \eql@shape@select
3063 }

```

`\eql@arrange@print@line` Select the appropriate adjustment method depending on the current alignment position, the selected tag placement if any: **TODO:** adjust

```

3064 \def\eql@arrange@print@line{%
3065   \eql@tagging@tagaddbox
3066   \csname eql@arrange%
3067     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3068     @init\endcsname
3069   \csname eql@arrange%
3070     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3071     @\ifdefined\eql@tagpos@reserve
3072       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3073       notag\fi\endcsname
3074 }

```

### J.3 Central Alignment

**TODO:** describe

```

3075 \def\eql@arrange@aligncenter@init{%
3076   \eql@tagging@aligncenter
3077   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3078     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3079 }

```

**TODO:** describe

```

3080 \def\eql@arrange@aligncenter@notag{%
3081   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3082     \ifdim\eql@line@offset@<\eql@marginleft@min@
3083       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3084     \else
3085       \eql@line@offset@
3086     \fi

```

```

3087 \eql@arrange@print@aligncenter\eql@line@offset@
3088 \else
3089 \ifdim\eql@line@offset@<\eql@marginleft@min@
3090 \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3091 \else
3092 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3093 \fi
3094 \fi
3095 }

```

**TODO:** describe

```

3096 \def\eql@arrange@aligncenter@tagsright{%
3097 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3098 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3099 \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3100 \else
3101 \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
3102 \fi
3103 \eql@arrange@print@aligncenter\eql@line@offset@
3104 \else
3105 \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
3106 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3107 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3108 \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3109 \else
3110 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3111 \fi
3112 \else
3113 \let\eql@tagpos@reserve\eql@false
3114 \eql@arrange@aligncenter@notag
3115 \fi
3116 \fi
3117 }

```

```

3118 \def\eql@arrange@aligncenter@tagsleft{%
3119 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3120 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3121 \ifdim\eql@line@offset@<\eql@tagwidth@
3122 \dimexpr\tw@\eql@tagwidth@-\eql@line@offset@\relax
3123 \else
3124 \eql@line@offset@
3125 \fi
3126 \eql@arrange@print@aligncenter\eql@line@offset@
3127 \else
3128 \eql@arrange@try\eql@tagwidth@
3129 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3130 \ifdim\eql@line@offset@<\eql@tagwidth@
3131 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3132 \else
3133 \eql@arrange@print@alignright\eql@tagwidth@\z@
3134 \fi
3135 \else
3136 \let\eql@tagpos@reserve\eql@false
3137 \eql@arrange@aligncenter@notag
3138 \fi
3139 \fi
3140 \else
3141 \eql@arrange@aligncenter@notag

```

```

3142 \fi
3143 }

```

## J.4 Left Alignment

```

3144 \def\eql@arrange@alignleft@init{%
3145   \eql@tagging@alignleft
3146   \eql@line@offset@dimexpr\eql@marginleft@+\eql@shape@amount@\relax
3147   \ifdim\eql@line@offset@<\eql@marginleft@min@
3148     \eql@line@offset@\eql@marginleft@min@
3149   \fi
3150 }

3151 \def\eql@arrange@alignleft@notag{%
3152   \ifdim\eql@line@offset@>\eql@marginleft@min@
3153     \eql@arrange@try\eql@line@offset@
3154     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3155       \eql@arrange@print@alignleft\eql@line@offset@\z@
3156     \else
3157       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3158     \fi
3159   \else
3160     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3161   \fi
3162 }

3163 \def\eql@arrange@alignleft@tagsright{%
3164   \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3165   \ifnum\eql@arrange@badness@<\eql@alignbadness@
3166     \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3167   \else
3168     \ifdim\eql@line@offset@>\eql@marginleft@min@
3169       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3170     \fi
3171     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3172       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3173     \else
3174       \let\eql@tagpos@reserve\eql@false
3175       \eql@arrange@alignleft@notag
3176     \fi
3177   \fi
3178 }

3179 \def\eql@arrange@alignleft@tagsleft{%
3180   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3181     \ifdim\eql@line@offset@>\eql@tagwidth@
3182       \eql@arrange@try\eql@line@offset@
3183       \ifnum\eql@arrange@badness@<\eql@alignbadness@
3184         \eql@arrange@print@alignleft\eql@line@offset@\z@
3185       \else
3186         \eql@arrange@try\eql@tagwidth@
3187         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3188           \eql@arrange@print@alignright\eql@tagwidth@\z@
3189         \else
3190           \let\eql@tagpos@reserve\eql@false
3191           \eql@arrange@print@alignright\eql@marginleft@min@\z@
3192         \fi
3193       \fi
3194   \else
3195     \eql@arrange@try\eql@tagwidth@

```

```

3196     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3197     \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3198   \else
3199     \let\eqL@tagpos@reserve\eqL@false
3200     \eqL@arrange@alignleft@notag
3201   \fi
3202 \fi
3203 \else
3204   \eqL@arrange@alignleft@notag
3205 \fi
3206 }

```

## J.5 Right Alignment

```

3207 \def\eqL@arrange@alignright@init{%
3208   \eqL@tagging@alignright
3209   \eqL@line@offset@dimexpr\eqL@marginright@-\eqL@shape@amount@relax
3210   \ifdim\eqL@line@offset@<z@
3211     \eqL@line@offset@z@
3212   \fi
3213 }

```

**TODO:** describe

```

3214 \def\eqL@arrange@alignright@notag{%
3215   \ifdim\eqL@line@offset@>z@
3216     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3217     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3218       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3219     \else
3220       \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3221     \fi
3222   \else
3223     \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3224   \fi
3225 }

```

**TODO:** describe

```

3226 \def\eqL@arrange@alignright@tagsright{%
3227   \ifdim\eqL@line@offset@>\eqL@tagwidth@
3228     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3229     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3230       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3231     \else
3232       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3233       \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3234         \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
3235       \else
3236         \let\eqL@tagpos@reserve\eqL@false
3237         \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3238       \fi
3239     \fi
3240   \else
3241     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3242     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3243       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3244     \else
3245       \let\eqL@tagpos@reserve\eqL@false
3246       \eqL@arrange@alignright@notag
3247     \fi

```

```

3248 \fi
3249 }

```

**TODO:** describe

```

3250 \def\eql@arrange@alignright@tagsleft{%
3251 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3252 \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3253 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3254 \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3255 \else
3256 \ifdim\eql@line@offset@>\z@
3257 \eql@arrange@try\eql@tagwidth@
3258 \fi
3259 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3260 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3261 \else
3262 \let\eql@tagpos@reserve\eql@false
3263 \eql@arrange@alignright@notag
3264 \fi
3265 \fi
3266 \else
3267 \eql@arrange@alignright@notag
3268 \fi
3269 }

```

## K Equations Box Environment

**TODO:** outline sequence of calls

**TODO:** describe

**TODO:** fixed width version (works only towards intercolumn stretch)?

**TODO:** vspace?!

### K.1 Line Breaks

`\eql@box@cr`

```

3270 \protected\def\eql@box@cr{%
3271 \eql@ampprotecttwo{\eql@ifnextchar@tight[]\eql@box@cr@skip\eql@box@cr@
3272 }
3273 \def\eql@box@cr@{%
3274 \eql@punct@apply@line
3275 \eql@hook@lineout
3276 \eql@box@lastcell
3277 \cr
3278 }
3279 \def\eql@box@cr@skip[#1]{%
3280 \eql@box@cr@
3281 \noalign{%
3282 \vskip\glueexpr#1\relax
3283 }%
3284 }

```

### K.2 Stacked Mode

```

3285 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}

```

**TODO:** templates

```

3286 \def\eql@box@open@stacked{%
3287   \eql@shape@align@enable
3288   \let\eql@box@lastcell\eql@box@lastcell@stacked
3289   \everycr{\noalign{%
3290 (dev)\eql@dev{starting line \the\eql@row@}%
3291     \global\advance\eql@row@\@ne
3292   }}%
3293   \tabskip\z@skip
3294   \halign\bgroup
3295     &%
3296     \global\let\eql@cell@container\@empty
3297     \setbox\eql@cellbox@\hbox{%
3298       \eql@strut@cell
3299       \@lign
3300       $\m@th\displaystyle
3301       \eql@hook@colin
3302       ##%
3303       \eql@punct@apply@col
3304       \eql@hook@colout
3305       \eql@tagging@mathsave
3306       $%
3307       \eql@tagging@mathaddlast
3308     }%
3309     \ifdefined\eql@shape@lastrow
3310       \eql@totalrows@\eql@row@
3311     \fi
3312     \eql@shape@eval
3313     \eql@cell@container
3314     \ifdefined\eql@frame@cmd
3315       \ifcase\eql@shape@pos@
3316         \eql@frame@measure
3317         \advance\eql@shape@amount@-\eql@frame@margin@
3318       \or\or
3319         \eql@frame@measure
3320         \advance\eql@shape@amount@+\eql@frame@margin@
3321       \fi
3322       \eql@frame@print
3323     \fi
3324     \ifcase\eql@shape@pos@
3325       \kern\eql@shape@amount@
3326       \box\eql@cellbox@
3327       \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3328       -\eql@shape@amount@+\@flushglue\relax
3329       \eql@tagging@alignleft
3330     \or
3331       \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3332       \box\eql@cellbox@
3333       \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3334       \eql@tagging@aligncenter
3335     \or
3336       \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3337       +\eql@shape@amount@+\@flushglue\relax
3338       \box\eql@cellbox@
3339       \kern-\eql@shape@amount@
3340       \eql@tagging@alignright
3341     \fi
3342     \tabskip\eql@colsep@\relax

```

```

3343 \crrr
3344 \noalign{%
3345 \global\let\eq\@shape@lastrow\eq\@false
3346 \eq\@hook@blockbefore
3347 }%
3348 \eq\@hook@blockin
3349 }
3350 \def\eq\@mode@stacked{\let\eq\@box@open\eq\@box@open@stacked}

```

### K.3 Aligned Mode

```

3351 \def\eq\@box@lastcell@odd{%
3352 &\omit
3353 \eq\@prevwidth@\wd\eq\@cellbox@
3354 \let\eq\@frame@cmd\eq\@frame@prevcmd
3355 \ifdefined\eq\@frame@cmd
3356 \eq\@frame@measure
3357 \advance\eq\@prevwidth@\eq\@frame@margin@
3358 \eq\@frame@print
3359 \fi
3360 \kern-\eq\@prevwidth@
3361 \unhbox\eq\@cellbox@
3362 \hfil
3363 &\omit\kern-\eq\@colsep@
3364 }%
3365 \def\eq\@box@lastcell@even{&\omit\kern-\eq\@colsep@}

3366 \def\eq\@box@open@aligned{%
3367 % \TODO templates
3368 \eq\@shape@align@disable
3369 \let\eq\@box@lastcell\@empty
3370 \everycr{\noalign{%
3371 (dev)\eq\@dev{starting new line}%
3372 }}%
3373 \tabskip\z@skip
3374 \halign\bgroup
3375 &%
3376 \let\eq\@box@lastcell\eq\@box@lastcell@odd
3377 \global\let\eq\@cell@container\@empty
3378 \global\setbox\eq\@cellbox@\hbox{%
3379 \eq\@strut@cell
3380 \@lign
3381 $\m@th\displaystyle
3382 \eq\@hook@colin
3383 ##%
3384 \eq\@class@innerleft
3385 \eq\@hook@innerleft
3386 \eq\@tagging@mathsave
3387 $%
3388 \eq\@tagging@mathaddlast
3389 }%
3390 \eq\@cell@container
3391 \hfil
3392 \kern\wd\eq\@cellbox@
3393 \ifdefined\eq\@frame@cmd
3394 \eq\@frame@measure
3395 \kern\eq\@frame@margin@
3396 \fi
3397 \global\let\eq\@frame@prevcmd\eq\@frame@cmd

```

```

3398     \tabskip\z@skip
3399     &%
3400     \eql@prevwidth@\wd\eql@cellbox@
3401     \let\eql@box@lastcell\eql@box@lastcell@even
3402     \let\eql@frame@cmd\eql@frame@prevcmd
3403     \global\let\eql@cell@container\@empty
3404     \setbox\eql@cellbox@\hbox{%
3405         \unhbox\eql@cellbox@
3406         \eql@strut@cell
3407         \@lign
3408         $\m@th\displaystyle
3409         \eql@hook@innerright
3410         \eql@class@innerright@sel
3411         ##%
3412         \eql@punct@apply@col
3413         \eql@hook@colout
3414         \eql@tagging@mathsave
3415         $%
3416         \eql@tagging@mathaddlast
3417     }%
3418     \eql@cell@container
3419     \ifdefined\eql@frame@cmd
3420         \eql@frame@measure
3421         \advance\eql@prevwidth@\eql@frame@margin@
3422         \eql@frame@print
3423     \fi
3424     \kern-\eql@prevwidth@
3425     \unhbox\eql@cellbox@
3426     \hfil
3427     \tabskip\eql@colsep@\relax
3428     \crrr
3429     \noalign{%
3430         \eql@hook@blockbefore
3431     }%
3432     \eql@hook@blockin
3433 }
3434 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

## K.4 Main

```

3435 \let\eql@box@box\vcenter
3436 \let\eql@box@open\@undefined
3437 \let\eql@box@frame\@firstofone
3438 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

```

**TODO:** can we avoid setting `\eql@totalrows@` globally here? **TODO:** this is needed for escaping the box and then set the alignment **TODO:** maybe determine alignment within inner math?! **TODO:** difficulty: last line being known (for steps) only after all cells have been processed. Note: only works for single column anyway! we do not have to cater for more!

```

3439 \def\eql@box@close{%
3440     \ifvmode\else
3441         \global\let\eql@shape@lastrow\eql@true
3442         \eql@punct@apply@block
3443         \eql@box@cr@
3444     \fi
3445     \noalign{%
3446         \eql@hook@blockafter

```



```

3447     \global\let\eq@shape@lastrow\eq@false
3448   }%
3449   \eq@tagging@tablesaveinner
3450 \egroup
3451 }

```

\eq@box@vcenter

```

3452 \def\eq@box@vcenter#1{%
3453   \ifmmode
3454     \vcenter{#1}%
3455   \else
3456     $\m@th\vcenter{#1}$%
3457   \fi
3458 }

```

\eq@box@start

```

3459 \let\eq@box@endmath\eq@false
3460 \def\eq@box@start{%
3461   \relax
3462   \ifmmode
3463     \let\eq@box@endmath\eq@false
3464   \else
3465     \let\eq@box@endmath\eq@true
3466     \expandafter$%$
3467   \fi
3468   \eq@box@processopt
3469   \eq@stack@save@box
3470   \let\eq@frame@cmd\@undefined
3471   \let\eq@layoutleft\eq@false
3472   \eq@row@z@
3473   \eq@totalrows@\@M
3474   \eq@shape@select
3475   \setbox\z@\ifx\eq@box@box\vcenter
3476     \expandafter\vbox
3477   \else
3478     \expandafter\eq@box@box
3479   \fi\bgroup
3480   \eq@display@nest
3481   \let\\eq@box@cr
3482   \eq@spread@set
3483   \eq@strut@make
3484   \eq@box@open
3485 }

```

\eq@box@end

```

3486 \def\eq@box@end{%
3487   \eq@box@close
3488   \egroup
3489   \eq@box@frame{%
3490     \ifdefined\eq@display@marginleft
3491       \hskip\glueexpr\eq@display@marginleft\relax
3492     \fi
3493     \ifx\eq@box@box\vcenter
3494       \eq@box@vcenter{\unvbox\z@}%
3495     \else
3496       \box\z@

```

```

3497 \fi
3498 \eql@tagging@tableaddinner
3499 \ifdefined\eql@display@marginright
3500 \hskip\glueexpr\eql@display@marginright\relax
3501 \fi
3502 }%
3503 \eql@stack@restore
3504 \ifdefined\eql@box@endmath
3505 \expandafter$%$
3506 \fi
3507 }

```

## K.5 Environment

`equationsbox` (*env.*)

```

3508 \newenvironment{equationsbox}{%
3509 (dev)\eql@dev@enterenv
3510 \eql@ampprotect\eql@box@testall\eql@box@start
3511 }{%
3512 \eql@box@end
3513 (dev)\eql@dev@leaveenv
3514 }

3515 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3516 \def\eql@box@parseopt{%
3517 \ifx\eql@parseopt@token[%]
3518 \let\eql@parseopt@next\eql@parseopt@opt
3519 \fi
3520 \ifx\eql@parseopt@token=%
3521 \let\eql@parseopt@next\eql@parseopt@lines
3522 \fi
3523 \ifx\eql@parseopt@token|
3524 \let\eql@parseopt@next\eql@parseopt@columns
3525 \fi
3526 }

```

`\eql@box@processopt` **TODO:** describe

```

3527 \def\eql@box@processopt{%
3528 \let\eql@box@frame\@firstofone
3529 \let\eql@display@marginleft\@undefined
3530 \let\eql@display@marginright\@undefined
3531 \eql@nextopt@process{equationsbox}%
3532 \let\eql@punct@block\eql@punct@main
3533 \let\eql@punct@main\relax
3534 \eql@colsep@\glueexpr\eql@box@colsep\relax
3535 \ifdefined\eql@paddingleft@val
3536 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3537 \else
3538 \eql@paddingleft@\z@
3539 \fi
3540 \ifdefined\eql@paddingright@val
3541 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3542 \else
3543 \eql@paddingright@\z@
3544 \fi
3545 \eql@indent@\glueexpr\eql@indent@val\relax
3546 }

```

## L Single-Line Equation

**TODO:** describe

### L.1 Native Mode

```
3547 \def\eql@single@start@native{%
3548   \eql@display@init
3549   \eql@display@print
3550   \let\raisetag\eql@raisetag@default
3551   \eql@shape@align@disable
3552   \eql@hook@eqin
3553 %   \mathopen{}%
3554 }%
```

**TODO:** describe

```
3555 \def\eql@single@end@native{%
3556 %   \mathclose{}%
3557   \eql@tags@container
3558   \eql@numbering@single@eval
3559   \if@eqnsw
3560     \ifdefined\eql@tagsleft
3561       \leqno
3562     \else
3563       \eqno
3564     \fi
3565     \eql@composetag@print
3566   \fi
3567   \eql@interline@container
3568   \advance\eql@belowspace@\eql@vspaceskip@
3569   \eql@display@container
3570   \eql@display@penalty
3571   \eql@display@vspace@native
3572 }%
```

### L.2 Print

```
3573 \def\eql@single@start@print{%
3574   \eql@display@init
3575   \eql@display@print
3576   \eql@shape@align@enable
3577   \eql@totalrows@\@ne
3578   \eql@row@\@ne
3579   \eql@arrange@init
3580   \global\let\eql@cell@container\@empty
3581   \prevgraf\numexpr\prevgraf+\@ne\relax
3582   \setbox\eql@cellbox@\hbox\bgroup
3583     \eql@restore@hfuzz
3584     \eql@strut@cell
3585     $\m@th\displaystyle%$
3586     \eql@hook@eqin
3587 }
3588 \def\eql@single@end@print{%
3589   \eql@tagging@mathsave
3590   $%$
3591   \hfil
3592   \kern\z@
```

```

3593 \egroup
3594 \prevgraf\numexpr\prevgraf-\@ne\relax

3595 \eq@shape@eval
3596 \eq@cell@container

3597 \ifdefined\eq@frame@cmd
3598 \eq@frame@adjust
3599 \fi

3600 \eq@cellwidth@\wd\eq@cellbox@
3601 \eq@line@height@\ht\eq@cellbox@
3602 \eq@line@depth@\dp\eq@cellbox@
3603 \eq@totalwidth@\eq@cellwidth@
3604 \eq@totalheight@\dimexpr\eq@line@height@+\eq@line@depth@\relax
3605 \eq@topheight@\eq@line@height@
3606 \eq@bottomdepth@\eq@line@depth@

3607 \eq@tags@container
3608 \eq@numbering@single@eval
3609 \if@eqnsw
3610 \eq@tagbox@make\eq@composetag@print
3611 \eq@tagrows@\@ne
3612 \ifdefined\eq@tagpos@reserve\else
3613 \eq@tagwidth@\z@
3614 \fi
3615 \eq@tagheight@block@\ht\eq@tagbox@
3616 \eq@tagdepth@block@\dp\eq@tagbox@
3617 \else
3618 \eq@numbering@warnunused
3619 \eq@tagwidth@\z@
3620 \eq@tagrows@\z@
3621 \fi
3622 \eq@tagwidth@max@\eq@tagwidth@
3623 \eq@tagpos@single@eval
3624 \eq@tagpos@print@line@eval

3625 \eq@intercolumns@\z@
3626 \eq@adjust@calc@lines

3627 \eq@display@halign@init{}%
3628 \halign{##\crr
3629 \noalign{\eq@display@halign@start}%
3630 \eq@arrange@print@line
3631 \cr
3632 \noalign{\eq@display@halign@end}%
3633 \eq@tagging@tablesavelines
3634 }%
3635 \eq@tagpos@print@line@end
3636 \eq@display@close
3637 }

```

## M Multi-Line with Single Column

**TODO:** outline sequence of calls

### M.1 Measure

**TODO:** describe

```

3638 \def\eql@lines@measure@line@begin{%
3639 (dev)\eql@dev{starting line \the\eql@row}%
3640 \eql@numbering@measure@line@begin
3641 \eql@hook@linein
3642 }

```

**TODO:** describe

```

3643 \def\eql@lines@measure@line@end{%
3644 \eql@punct@apply@line
3645 \eql@hook@lineout
3646 }

```

**TODO:** describe **TODO:** it would be an option to add the absolute shove amount to the calculation of the maximum width

```

3647 \def\eql@lines@measure@cell{%
3648 \ifdefined\eql@frame@cmd
3649 \ifcase\eql@shape@pos@
3650 \eql@frame@measure
3651 \advance\eql@shape@amount@-\eql@frame@margin@
3652 \or\or
3653 \eql@frame@measure
3654 \advance\eql@shape@amount@+\eql@frame@margin@
3655 \fi
3656 \eql@frame@print
3657 \fi
3658 \eql@cellwidth@\wd\eql@cellbox@
3659 \eql@line@height@\ht\eql@cellbox@
3660 \eql@line@depth@\dp\eql@cellbox@
3661 \eql@dimensions@startrow
3662 \eql@dimensions@savecell
3663 \kern\eql@cellwidth@
3664 }

```

**\eql@lines@measure**

```

3665 \def\eql@lines@measure{%
3666 (dev)\eql@dev@enter\eql@lines@measure
3667 \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3668 \eql@totalrows@\@M
3669 \eql@shape@select

3670 \setbox\z@\vbox{\measuring@true\halign{%
3671 \global\let\eql@cell@container\@empty
3672 \setbox\eql@cellbox@\hbox{%
3673 \eql@strut@cell
3674 \@lign
3675 $\m@th\displaystyle
3676 \eql@hook@colin
3677 ##%
3678 \eql@punct@apply@col
3679 \eql@hook@colout
3680 $%
3681 }%
3682 \ifdefined\eql@shape@lastrow
3683 \eql@totalrows@\eql@row@
3684 \fi
3685 \eql@shape@eval
3686 \eql@cell@container

```

```

3687 \eql@lines@measure@cell
3688 \eql@measure@tag
3689 \eql@measure@endrow
3690 \crrr

3691 \noalign{%
3692 \global\let\eql@shape@lastrow\eql@false
3693 \eql@hook@blockbefore
3694 }%
3695 \eql@hook@blockin
3696 \eql@scan@body
3697 \ifvmode\else
3698 \global\let\eql@shape@lastrow\eql@true
3699 \eql@punct@apply@block
3700 \eql@hook@blockout
3701 \eql@display@endline
3702 \cr
3703 \fi
3704 \omit
3705 \cr
3706 \noalign{%
3707 \eql@hook@blockafter
3708 \global\let\eql@shape@lastrow\eql@false
3709 }%
3710 }%

3711 \eql@measure@close

3712 \setbox\z@\vbox{%
3713 \unvbox\z@
3714 \unpenalty
3715 \global\setbox\@ne\lastbox
3716 }%
3717 \eql@totalwidth@\wd\@ne

3718 (dev)\eql@dev@leave\eql@lines@measure
3719 }

```

## M.2 Column Placement

**TODO:** describe Find the best row for tag placement:

```

3720 \def\eql@lines@adjust{%
3721 \eql@tagpos@adjust@eval
3722 \eql@adjust@calc@lines
3723 \eql@numbering@best@eval
3724 }

```

## M.3 Print

**TODO:** describe

mes@print@line@begin

```

3725 \def\eql@lines@print@line@begin{%
3726 (dev)\eql@dev{starting line \the\eql@row}%
3727 \eql@numbering@print@line@begin
3728 \eql@hook@linein

```

3729 }

**TODO:** describe

```
3730 \def\eq@lines@print@line@end{%
3731   \eq@punct@apply@line
3732   \eq@hook@lineout
3733 }
```

**TODO:** describe

```
3734 \def\eq@lines@print@line@adjust{%
3735   \ifdefined\eq@frame@cmd
3736     \ifcase\eq@shape@pos@
3737       \eq@frame@measure
3738       \advance\eq@shape@amount@-\eq@frame@margin@
3739     \or\or
3740       \eq@frame@measure
3741       \advance\eq@shape@amount@+\eq@frame@margin@
3742     \fi
3743     \eq@frame@adjust
3744   \fi
3745   \eq@cellwidth@\wd\eq@cellbox@
3746   \eq@line@height@\ht\eq@cellbox@
3747   \eq@line@depth@\dp\eq@cellbox@
3748   \eq@numbering@print@line@eval
3749   \if@eqnsw
3750     \eq@tagbox@make\eq@composetag@print
3751   \fi
3752   \eq@tagpos@print@line@eval
3753   \eq@arrange@print@line
3754   \eq@tagpos@print@line@end
3755 }
```

**TODO:** describe

```
3756 \def\eq@lines@print{%
3757 (dev)\eq@dev@center\eq@lines@print
3758   \eq@arrange@init
3759   \eq@display@halign@init\eq@lines@print@line@begin
3760   \eq@display@halign@letcr\eq@lines@print@line@end
3761   \tabskip\z@skip

3762   \halign{%
3763     \global\let\eq@cell@container\@empty
3764     \setbox\eq@cellbox@\hbox{%
3765       \eq@restore@hfuzz
3766       \eq@strut@cell
3767       \@lign
3768       $\m@th\displaystyle
3769       \eq@hook@colin
3770       ##%
3771       \eq@punct@apply@col
3772       \eq@hook@colout
3773       \eq@tagging@mathsave
3774       $%
3775       \hfil
3776       \kern\z@
3777     }%
3778     \eq@shape@eval
3779     \eq@cell@container
```

```

3780     \eql@lines@print@line@adjust
3781 \crrr

3782 \noalign{%
3783     \eql@display@halign@start
3784     \eql@numbering@print@block@begin
3785     \eql@hook@blockbefore
3786 }%
3787 \eql@hook@blockin
3788 \eql@scan@body
3789 \ifvmode\else
3790     \relax
3791     \eql@punct@apply@block
3792     \eql@hook@blockout
3793     \eql@display@endline
3794 \cr
3795 \fi
3796 \noalign{%
3797     \eql@hook@blockafter
3798     \eql@display@halign@end
3799 (dev)\eql@dev@leave\eql@lines@print
3800 }%
3801 \eql@tagging@tablesavelines
3802 }%
3803 }

```

## N Multi-Line with Multiple Columns

**TODO:** describe **TODO:** outline sequence of calls

### N.1 Support

**TODO:** describe

```

\eql@columns@add@amp
@columns@completerow
3804 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3805 \def\eql@columns@completerow{%
3806     \count@ \numexpr\eql@totalcolumns@+\@ne-\eql@column@ \relax
3807     \edef\eql@tmp{%
3808         \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3809     \eql@tmp
3810 }

3811 \def\eql@columns@overfull{%
3812     \dimen@ \eql@line@width@
3813     \advance\dimen@-\hfuzz
3814     \ifdim\dimen@>\displaywidth
3815         \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3816         \wd\z@\z@
3817         \ht\z@\eql@line@height@
3818         \dp\z@\eql@line@depth@
3819         \box\z@
3820     \fi
3821 }

```



## N.2 Transpose

**TODO:** describe

**TODO:** describe

```
3822 \let\eql@transpose@active\eql@false
3823 \def\eql@transpose@end{\eql@transpose@end}
3824 \def\eql@transpose@skip{&\eqnpunct{}}
3825 \def\eql@transpose@complete{%
3826   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}
```

**TODO:** describe

```
3827 \def\eql@transpose{%
3828   \eql@totalcolumns@z@
3829   \eql@totalrows@z@
3830   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
3831   \eql@scan@reg@{}%
3832   \eql@row@z@
3833   \eql@transpose@output@row
3834 }
```

**TODO:** describe

```
3835 \def\eql@transpose@save@col#1{%
3836   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
3837     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}
```

**TODO:** describe

```
3838 \def\eql@transpose@scan@col#1&{%
3839   \def\eql@tmpa{#1}%
3840   \ifx\eql@tmpa\eql@transpose@end\else
3841     \advance\eql@totalcolumns@\@ne
3842     \eql@row@z@
3843     \let\eql@transpose@data@col@\empty
3844     \eql@transpose@scan@row#1\\eql@transpose@end\\%
3845     \ifnum\eql@row@>\eql@totalrows@
3846       \eql@totalrows@\eql@row@
3847     \fi
3848     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
3849     \expandafter\eql@transpose@scan@col
3850   \fi
3851 }
```

**TODO:** describe

```
3852 \def\eql@transpose@append@row#1{%
3853   \advance\eql@row@\@ne
3854   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}
```

**TODO:** describe

```
3855 \def\eql@transpose@scan@row#1\\{%
3856   \def\eql@tmpa{#1}%
3857   \ifx\eql@tmpa\eql@transpose@end\else
3858     \ifx\eql@transpose@active+
3859       \eql@transpose@scan@cell#1&\eql@transpose@end&%
3860     \else
3861       \eql@transpose@append@row{#1}%
3862     \fi
```

```

3863 \expandafter\eql@transpose@scan@row
3864 \fi
3865 }

```

**TODO:** describe

```

3866 \def\eql@transpose@scan@cell#1&#2&{%
3867 \def\eql@tmpa{#2}%
3868 \ifx\eql@tmpa\eql@transpose@end
3869 \eql@transpose@append@row{#1}%
3870 \else
3871 \eql@transpose@append@row{#1&#2}%
3872 \expandafter\eql@transpose@scan@cell@next
3873 \fi
3874 }

```

**TODO:** describe

```

3875 \def\eql@transpose@scan@cell@next#1&{%
3876 \def\eql@tmpa{#1}%
3877 \ifx\eql@tmpa\eql@transpose@end\else
3878 \eql@transpose@append@row{&#1}%
3879 \expandafter\eql@transpose@scan@cell@next
3880 \fi
3881 }

```

**TODO:** describe

```

3882 \def\eql@transpose@output@row{%
3883 \ifnum\eql@row@<\eql@totalrows@
3884 \advance\eql@row@\@ne
3885 \eql@column@\z@
3886 \eql@transpose@output@col
3887 \ifnum\eql@row@<\eql@totalrows@
3888 \eql@scan@addto\\%
3889 \fi
3890 \expandafter\eql@transpose@output@row
3891 \fi
3892 }

```

**TODO:** describe

```

3893 \def\eql@transpose@output@col{%
3894 \ifnum\eql@column@<\eql@totalcolumns@
3895 \advance\eql@column@\@ne
3896 \csname eql@transpose@data@col@\the\eql@column@\endcsname
3897 \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
3898 \ifnum\eql@column@<\eql@totalcolumns@
3899 \eql@scan@addto{\eql@transpose@complete}%
3900 \fi
3901 \expandafter\eql@transpose@output@col
3902 \fi
3903 }

```

### N.3 Measure

**TODO:** describe **TODO:** this is called also for extra line and concluding cr

s@measure@line@begin

```

3904 \def\eql@columns@measure@line@begin{%
3905 (dev)\eql@dev{starting line \the\eql@row}%
3906   \global\eql@column@\z@
3907   \global\eql@line@height@\z@
3908   \global\eql@line@depth@\z@
3909   \eql@numbering@measure@line@begin
3910   \eql@hook@linein
3911 }

3912 \def\eql@columns@measure@cell{%
3913   \eql@cellwidth@\wd\eql@cellbox@
3914   \ifdefined\eql@frame@cmd
3915     \eql@frame@measure
3916     \advance\eql@cellwidth@\eql@frame@margin@
3917   \fi
3918   \ifdim\ht\eql@cellbox@>\eql@line@height@
3919     \global\eql@line@height@\ht\eql@cellbox@
3920   \fi
3921   \ifdim\dp\eql@cellbox@>\eql@line@depth@
3922     \global\eql@line@depth@\dp\eql@cellbox@
3923   \fi
3924   \ifnum\eql@column@=\@ne
3925     \eql@dimensions@startrow
3926   \fi
3927   \ifodd\eql@column@
3928     \eql@shape@pos@\tw@
3929   \else
3930     \eql@shape@pos@\z@
3931   \fi
3932   \eql@shape@amount@\z@
3933   \eql@dimensions@savercell
3934   \ifodd\eql@column@\else
3935     \eql@dimensions@savesep
3936   \fi
3937   \kern\eql@cellwidth@
3938 }

```

mns@measure@line@end

```

3939 \def\eql@columns@measure@line@end{%
3940   \eql@punct@apply@line
3941   \eql@hook@lineout
3942   &\omit
3943   \ifnum\eql@column@>\eql@totalcolumns@
3944     \global\eql@totalcolumns@\eql@column@
3945   \fi

```

**TODO:** not sure whether saving the last cell value makes sense, but rather not increase `\eql@totalcolumns@` because that will disable the fallback to lines mode. **TODO:** additional column in width table is accounted for in column table

```

3946   \ifdefined\eql@frame@cmd
3947     \advance\eql@column@\@ne
3948     \wd\eql@cellbox@\z@
3949     \eql@columns@measure@cell
3950   \fi
3951   \eql@measure@tag
3952   \eql@measure@endrow
3953 }

```

\eql@columns@measure

```
3954 \def\eql@columns@measure{%
3955 \dev\eql@dev@center\eql@columns@measure
3956 \eql@totalcolumns@z@
3957 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end

3958 \setbox\z@\vbox{\measuring@true\halign{%
3959   &%
3960   \global\advance\eql@column@ \@ne
3961   \global\let\eql@cell@container\@empty
3962   \global\setbox\eql@cellbox@\hbox{%
3963     \eql@strut@cell
3964     \@lign
3965     $\m@th\displaystyle
3966     \eql@hook@colin
3967     ##%
3968     \eql@class@innerleft
3969     \eql@hook@innerleft
3970     $%
3971   }%
3972   \eql@cell@container
3973   \hfil
3974   \eql@columns@measure@cell
3975   \global\let\eql@frame@prevcmd\eql@frame@cmd
3976   &%
3977   \eql@prevwidth@\wd\eql@cellbox@
3978   \let\eql@frame@cmd\eql@frame@prevcmd
3979   \global\advance\eql@column@ \@ne
3980   \global\let\eql@cell@container\@empty
3981   \setbox\eql@cellbox@\hbox{%
3982     \eql@strut@cell
3983     \@lign
3984     $\m@th\displaystyle
3985     \eql@hook@innerright
3986     \eql@class@innerright@sel
3987     ##%
3988     \eql@punct@apply@col
3989     \eql@hook@colout
3990     $%
3991   }%
3992   \eql@cell@container
3993   \eql@columns@measure@cell
3994   \hfil
3995   \crrr

3996   \noalign{%
3997     \eql@hook@blockbefore
3998   }%
3999   \eql@hook@blockin
4000   \eql@scan@body

4001   \ifvmode\else
4002     \eql@punct@apply@block
4003     \eql@hook@blockout
4004     \eql@display@endline
4005     \cr
4006   \fi
4007   \noalign{%
4008     \eql@hook@blockafter
```

```
4009 }%
```

**TODO:** note we also include the tag column as a backup

```
4010 \omit
4011 \eql@column@\@ne
4012 \eql@columns@completerow
4013 \cr
4014 }}%
```

```
4015 \eql@measure@close
```

```
4016 \setbox\z@\vbox{%
4017 \unvbox\z@
4018 \unpenalty
4019 \global\setbox\@ne\lastbox
4020 }%
4021 \eql@totalwidth@\wd\@ne
```

**TODO:** why not recycle box contents altogether?!

```
4022 \let\eql@colwidth@tab@empty
4023 \loop
4024 \setbox\@ne\hbox{%
4025 \unhbox\@ne
4026 \unskip
4027 \global\setbox\thr@@\lastbox
4028 }%
4029 \ifhbox\thr@@
4030 \eql@colwidth@save{\wd\thr@@}%
4031 \repeat

4032 (dev)\eql@dev@leave\eql@columns@measure
4033 }
```

## N.4 Columns Placement

**TODO:** describe Make sure we have complete pairs of right and left adjusted columns, otherwise add a final empty column:

```
4034 \def\eql@columns@adjust{%
4035 \ifodd\eql@totalcolumns@
4036 \advance\eql@totalcolumns@\@ne
4037 \fi
4038 \eql@tagpos@adjust@eval
4039 \eql@adjust@calc@columns
4040 }
```

## N.5 Print

**TODO:** describe

```
mns@print@line@begin
```

```
4041 \def\eql@columns@print@line@begin{%
4042 (dev)\eql@dev{starting line \the\eql@row}%
4043 \global\eql@column@\z@
4044 \global\eql@line@pos@\eql@marginleft@
```

```

4045 \global\eql@line@width@ \z@
4046 \global\eql@line@avail@ \eql@totalwidth@
4047 \global\eql@line@height@ \z@
4048 \global\eql@line@depth@ \z@
4049 \eql@numbering@print@line@begin
4050 \eql@hook@linein
4051 }

```

l@columns@print@cell

```

4052 \def\eql@columns@print@cell{%
4053 \eql@cellwidth@\wd\eql@cellbox@
4054 \ifodd\eql@column@
4055 \ifdefined\eql@frame@cmd
4056 \eql@frame@measure
4057 \advance\eql@cellwidth@\eql@frame@margin@
4058 \fi
4059 \dimen@\z@
4060 \else
4061 \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```

4062 \ifdefined\eql@frame@cmd
4063 \eql@frame@measure
4064 \advance\eql@cellwidth@\eql@frame@margin@
4065 \advance\eql@prevwidth@\eql@frame@margin@
4066 \eql@frame@print
4067 \fi

```

update height and depth

```

4068 \ifdim\ht\eql@cellbox@>\eql@line@height@
4069 \global\eql@line@height@\ht\eql@cellbox@
4070 \fi
4071 \ifdim\dp\eql@cellbox@>\eql@line@depth@
4072 \global\eql@line@depth@\dp\eql@cellbox@
4073 \fi

```

print box

```

4074 \kern-\eql@prevwidth@
4075 \unhbox\eql@cellbox@
4076 \dimen@-\eql@cellwidth@
4077 \fi

```

enforce given width: hopefully measure was correct, but need a precise width for tag placement

```

4078 \advance\dimen@\eql@colwidth@get\eql@column@\relax
4079 \kern\dimen@

```

update available and used space

```

4080 \dimen@\eql@colwidth@get\eql@column@\relax
4081 \ifdim\eql@cellwidth@>\z@
4082 \ifdim\eql@line@width@=\z@
4083 \eql@line@avail@\eql@line@pos@
4084 \ifodd\eql@column@
4085 \advance\eql@line@avail@\dimen@
4086 \advance\eql@line@avail@-\eql@cellwidth@
4087 \fi

```

```

4088     \global\eql@line@avail@\eql@line@avail@
4089     \fi
4090     \eql@line@width@\eql@line@pos@
4091     \ifodd\eql@column@
4092         \advance\eql@line@width@\dimen@
4093     \else
4094         \advance\eql@line@width@\eql@cellwidth@
4095     \fi
4096     \global\eql@line@width@\eql@line@width@
4097 \fi
4098 \advance\eql@line@pos@\dimen@
4099 \ifodd\eql@column@\else
4100     \advance\eql@line@pos@\eql@colsep@
4101 \fi
4102 \global\eql@line@pos@\eql@line@pos@
4103 }

4104 \def\eql@columns@print@trailright{%
4105     &\omit
4106     \eql@prevwidth@\wd\eql@cellbox@
4107     \let\eql@frame@cmd\eql@frame@prevcmd
4108     \global\advance\eql@column@\@ne
4109     \eql@columns@print@cell
4110 }

```

lums@print@line@end

```

4111 \def\eql@columns@print@line@end{%
4112     \eql@punct@apply@line
4113     \eql@hook@lineout
4114 % \TODO add an even column with empty stuff if box processing deferred
4115     \ifodd\eql@column@
4116         \expandafter\eql@columns@print@trailright
4117     \fi
4118     \eql@columns@completerow
4119     \eql@columns@print@tag
4120 }

```

ql@columns@print@tag

```

4121 \def\eql@columns@print@tag{%
4122     \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4123     \eql@display@firstavail@set\eql@line@avail@
4124     \eql@columns@overfull
4125     \eql@numbering@print@line@eval
4126     \if@eqnsw
4127         \eql@tagbox@make\eql@composetag@print
4128     \fi
4129     \eql@tagpos@print@line@eval
4130     \eql@tagbox@print@cell
4131     \eql@tagpos@print@line@end
4132 }

```

\eql@columns@print

```

4133 \def\eql@columns@print{%
4134     <dev>\eql@dev@enter\eql@columns@print

```

```

4135 \eq@shape@align@disable
4136 \eq@display@halign@init\eq@columns@print@line@begin
4137 \eq@display@halign@letcr\eq@columns@print@line@end
4138 \tabskip\eq@marginleft@

4139 \halign{%
4140   &%
4141   \global\advance\eq@column@ \@ne
4142   \global\let\eq@cell@container \@empty
4143   \global\setbox\eq@cellbox@ \hbox{%
4144     \eq@strut@cell
4145     \@lign
4146     $\m@th\displaystyle
4147     \eq@hook@colin
4148     ##%
4149     \eq@class@innerleft
4150     \eq@hook@innerleft
4151     \eq@tagging@mathsave
4152     $%
4153     \eq@tagging@mathaddlast
4154   }%
4155   \eq@cell@container
4156   \hfil
4157   \eq@columns@print@cell
4158   \global\let\eq@frame@prevcmd\eq@frame@cmd
4159   \tabskip\z@skip
4160   &%
4161   \eq@prevwidth@\wd\eq@cellbox@
4162   \let\eq@frame@cmd\eq@frame@prevcmd
4163   \global\advance\eq@column@ \@ne
4164   \global\let\eq@cell@container \@empty
4165   \setbox\eq@cellbox@ \hbox{%
4166     \unhbox\eq@cellbox@
4167     \eq@strut@cell
4168     \@lign
4169     $\m@th\displaystyle
4170     \eq@hook@innerright
4171     \eq@class@innerright@sel
4172     ##%
4173     \eq@punct@apply@col
4174     \eq@hook@colout
4175     \eq@tagging@mathsave
4176     $%
4177     \eq@tagging@mathaddlast
4178   }%
4179   \eq@cell@container
4180   \eq@columns@print@cell
4181   \hfil
4182   \tabskip\eq@colsep@\relax
4183 \crr

4184 \noalign{%
4185   \eq@display@halign@start
4186   \eq@numbering@print@block@begin
4187   \eq@hook@blockbefore
4188 }%
4189 \eq@hook@blockin
4190 \eq@scan@body
4191 \ifvmode\else

```



```

4192     \relax
4193     \eql@punct@apply@block
4194     \eql@hook@blockout
4195     \eql@display@endline
4196     \cr
4197     \fi
4198     \noalign{%
4199         \eql@hook@blockafter
4200         \eql@display@halign@end
4201     }%
4202     \dev\eql@dev@leave\eql@columns@print
4203     \eql@tagging@tables@savealign
4204     }%
4205 }

```

## O Interface

### O.1 Scanning the Equation Body

The multi-line equation environment must scan its body twice: once to determine how wide the columns are and then to actually typeset them. This means that we must collect all text in this body before calling the environment macros. The mechanism and its description follows `amsmath` closely.

#### Token Register.

`\eql@scan@reg@` We start by defining a token register to hold the equation body.

```
4206 \newtoks\eql@scan@reg@
```

`\eql@scan@body@dump` The macro `\eql@scan@body@dump` dumps the equation body from the register so that we do not have to pass it around in arguments. The macro `\eql@scan@body@rescan` rescans the tokens so that special commands such as `\verb` can be processed properly. The register `\eql@scan@body` holds the currently selected mode of operation:

```

4207 \def\eql@scan@body@dump{\the\eql@scan@reg@}
4208 \def\eql@scan@body@rescan{%
4209     \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
4210 \let\eql@scan@body\eql@scan@body@dump

```

`\eql@scan@addto` We define a macro to append to the token register `\eql@scan@reg@`:

```
4211 \long\def\eql@scan@addto#1{\eql@scan@reg@\expandafter{\the\eql@scan@reg@#1}}
```

**Environment Body.** The following mechanism scans the contents of an environment taking into account nested environments that may be contained in the body.

`\eql@scan@env` The macro `\eql@scan@env` starts the scan for the `\end{...}` command of the current environment. The argument is a call-back macro to process the body in `\eql@scan@reg@`:

```

4212 \def\eql@scan@env#1{%
4213     \dev\eql@dev@enter\eql@scan@env
4214     \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenvir}}%
4215     \eql@scan@reg@{\def\eql@scan@stack{b}}%

```

We call `\eql@scan@env@iterate` which will scan until the next occurrence of `\end` and then count the number of occurrences of `\begin` before `\end` in `\eql@scan@stack`. If we simply called `\eql@scan@env@iterate` directly, the error message for an unwanted `\par` token (usually from a blank line) would refer to `\eql@scan@env@iterate` which would not be illuminating. We use a little finesse to get a more intelligible error message: We use the actual environment name as the name of the temporary function that is `\let` to `\eql@scan@env@iterate`:

```

4216 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currentenv\endcsname}%
4217 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
4218 \ifdefined\eql@scan@par
4219 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
4220 \else
4221 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
4222 \fi
4223 \eql@scan@iterate
4224 }
```

`\eql@scan@env@iterate` `\eql@scan@env@iterate` takes two arguments: the first will consist of all text up to the next `\end` command, the second will be the `\end` command's argument. If there are any extra `\begin` commands in the body text, a marker is pushed onto a stack via `\eql@scan@env@count`. An empty state for this stack means that we have reached the `\end` that matches our original `\begin`. Otherwise we need to include the `\end` and its argument in the material that we are adding to our environment body accumulator:

```

4225 \long\def\eql@scan@env@iterate#1\end#2{%
4226 \edef\eql@scan@stack{%
4227 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4228 \ifx\@empty\eql@scan@stack
4229 \@checkend{#2}%
4230 \eql@scan@addto{#1}%
4231 \expandafter\let\eql@scan@iterate\eql@scan@env@org
4232 (dev)\eql@dev@leave\eql@scan@env
4233 \expandafter\eql@scan@end
4234 \else
4235 \eql@scan@addto{#1\end{#2}}%
4236 \expandafter\eql@scan@iterate
4237 \fi
4238 }
```

`\eql@scan@env@iterate@nopar` Version of `\eql@scan@env@iterate` which does not accept `\par` within the argument:

```

4239 \def\eql@scan@env@iterate@nopar#1\end#2{\eql@scan@env@iterate#1\end{#2}}
```

`\eql@scan@env@count` When adding a piece of the current environment's contents to `\eql@scan@reg@`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

4240 \long\def\eql@scan@env@count#1\begin#2{%
4241 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4242 }
```

The call-back macro `\eql@scan@env@cancel` ignores the body as well as the end clause for the environment:

```

4243 \def\eql@scan@env@cancel{%
4244 \@namedef{end\@currentenv}{\ignorespacesafterend}%
4245 }
```

**Square Brackets.** The following is a version of the above mechanism that scans for an equation body enclosed by `\[...]` paying attention to potential further instances of the square bracket enclosures contained in the body.

`\eql@scan@sqr` Start scanning for `\]`:

```

4246 \def\eql@scan@sqr#1{%
4247 (dev)\eql@dev@enter\eql@scan@sqr
4248 \def\eql@scan@end{#1\}}%
4249 \eql@scan@reg@{} \def\eql@scan@stack{b}%
4250 \let\eql@scan@sqr@org\[%\]
4251 \ifdefined\eql@scan@par
4252 \let\[\eql@scan@sqr@iterate%\]
4253 \else
4254 \let\[\eql@scan@sqr@iterate@nopar%\]
4255 \fi
4256 \[%\]
4257 }
```

`\eql@scan@sqr@iterate` Iterate until we find a balanced pairing of square brackets. Then call the call-back macro:

```

4258 \long\def\eql@scan@sqr@iterate#1\{%
4259 \edef\eql@scan@stack{%
4260 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4261 \ifx\@empty\eql@scan@stack
4262 \let\[\eql@scan@sqr@org%\]
4263 \eql@scan@addto{#1}%
4264 (dev)\eql@dev@leave\eql@scan@sqr
4265 \expandafter\eql@scan@end
4266 \else
4267 \eql@scan@addto{#1\}}%
4268 \expandafter\[%\]
4269 \fi
4270 }
```

`\eql@scan@sqr@iterate@nopar` Version of `\eql@scan@sqr@iterate` which does not accept `\par` within the argument:

```

4271 \def\eql@scan@sqr@iterate@nopar#1\{\eql@scan@sqr@iterate#1\}}
```

`\eql@scan@sqr@count` Push a ‘b’ for every encountered instance of ‘`\[`’:

```

4272 \long\def\eql@scan@sqr@count#1\[#2%\]
4273 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4274 }
```

`\eql@scan@sqrang@cancel` The call-back macro `\eql@scan@sqrang@cancel` ignores the body and the closing bracket:

```

4275 \def\eql@scan@sqrang@cancel{\expandafter\ignorespaces\@gobble}
```

**Angle Brackets.** The following is another version of the mechanism which scans for an equation body enclosed by `\<...>`.

`\eql@scan@ang` Start scanning for `\>`:

```

4276 \def\eql@scan@ang#1{%
4277 (dev)\eql@dev@enter\eql@scan@ang
4278 \def\eql@scan@end{#1\>}%
4279 \eql@scan@reg@{} \def\eql@scan@stack{b}%

```

```

4280 \let\eql@scan@ang@org\<%\>
4281 \ifdefined\eql@scan@par
4282   \let\<\eql@scan@ang@iterate%\>
4283 \else
4284   \let\<\eql@scan@ang@iterate@nopar%\>
4285 \fi
4286 \<%\>
4287 }

```

`\eql@scan@ang@iterate` Iterate until we find a balanced pairing of angle brackets:

```

4288 \long\def\eql@scan@ang@iterate#1\>{%
4289   \edef\eql@scan@stack{%
4290     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4291   \ifx\@empty\eql@scan@stack
4292     \let\<\eql@scan@ang@org%\>
4293     \eql@scan@addto{#1}%
4294   (dev)\eql@dev@leave\eql@scan@ang
4295     \expandafter\eql@scan@end
4296   \else
4297     \eql@scan@addto{#1\>}%
4298     \expandafter\<%\>
4299   \fi
4300 }

```

`\an@ang@iterate@nopar` Version of `\eql@scan@ang@iterate` which does not accept `\par` within the argument:

```

4301 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}

```

`\eql@scan@ang@count` Push a ‘b’ for every encountered instance of ‘<’:

```

4302 \long\def\eql@scan@ang@count#1\<#2{%\>
4303   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4304 }

```

## O.2 Options Processing

`\eqlequations@testall` The macro sequence started by `\eqlequations@testall` scans for optional arguments to the equation environments and appends them to the argument list using `\eqnaddopt`. All arguments are scanned such that any spaces stop the scanning and such that any alignment markers ‘&’ cannot interfere: **TODO:** update

```

4305 \def\eqlequations@testall{\eql@parseopt\eqlequations@parseopt}
4306 \def\eqlequations@parseopt{%
4307   \ifx\eql@parseopt@token*%
4308     \let\eql@parseopt@next\eql@parseopt@nonumber
4309   \fi
4310   \ifx\eql@parseopt@token!%
4311     \let\eql@parseopt@next\eql@parseopt@donumber
4312   \fi
4313   \ifx\eql@parseopt@token/%
4314     \let\eql@parseopt@next\eql@parseopt@transpose
4315   \fi
4316   \ifx\eql@parseopt@token[%
4317     \let\eql@parseopt@next\eql@parseopt@opt
4318   \fi
4319   \ifx\eql@parseopt@token\eql@atxi
4320     \let\eql@parseopt@next\eql@parseopt@label

```

```

4321 \fi
4322 \ifx\eql@parseopt@token\eql@atxii
4323   \let\eql@parseopt@next\eql@parseopt@label
4324 \fi
4325 \ifx\eql@parseopt@token.%
4326   \let\eql@parseopt@next\eql@parseopt@punctdot
4327 \fi
4328 \ifx\eql@parseopt@token,%
4329   \let\eql@parseopt@next\eql@parseopt@punctcomma
4330 \fi
4331 \ifx\eql@parseopt@token~%
4332   \let\eql@parseopt@next\eql@parseopt@punctoff
4333 \fi
4334 \ifx\eql@parseopt@token'
4335   \let\eql@parseopt@next\eql@parseopt@punctall
4336 \fi
4337 \ifx\eql@parseopt@token-%
4338   \let\eql@parseopt@next\eql@parseopt@single
4339 \fi
4340 \ifx\eql@parseopt@token=%
4341   \let\eql@parseopt@next\eql@parseopt@lines
4342 \fi
4343 \ifx\eql@parseopt@token|
4344   \let\eql@parseopt@next\eql@parseopt@columns
4345 \fi
4346 \ifx\eql@parseopt@token\label
4347   \let\eql@parseopt@next\eql@parseopt@end
4348 \fi
4349 \ifx\eql@parseopt@token\begin
4350   \let\eql@parseopt@next\eql@parseopt@end
4351 \fi
4352 }

```

**equations@processopt** The macro `\eql@equations@processopt` processes the options received by `\eqnaddopt`. First, clear several non-persistent registers (labels, tags, direct vertical spacing). Then process the arguments. Finally evaluate `\eql@indent@val` and `\eql@tagsepmin@val` and prevent main punctuation from being passed to nested environments:

```

4353 \def\eql@equations@processopt{%
4354   \let\eql@tags@container@block\eql@tags@container@clear
4355   \let\eql@tags@frame@cmd\@firstofone
4356   \let\eql@skip@force@above\@undefined
4357   \let\eql@skip@force@below\@undefined
4358   \let\eql@skip@force@leave\@undefined
4359   \let\eql@display@linewidth\@undefined
4360   \let\eql@display@marginleft\@undefined
4361   \let\eql@display@marginright\@undefined
4362   \eql@abovespace@\z@skip
4363   \eql@belowspace@\z@skip
4364   \eql@displaybreak@prepen@\@MM
4365   \eql@displaybreak@postpen@\@MM
4366   \eql@nextopt@process{equations}%
4367   \let\eql@punct@block\eql@punct@main
4368   \let\eql@punct@main\relax
4369   \eql@indent@\glueexpr\eql@indent@val\relax
4370   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4371 }

```

### O.3 Single-Line Main

In the following, we define the main routine for the single-line equation mode.

`\eql@single@cr` Cannot use line breaks, produce an error message:

```
4372 \def\eql@single@cr{%
4373   \eql@error{Cannot use '\string\\' within display equation.
4374     Please switch to equations environment}}%
4375 }
```

`\eql@single@start` Opening code for single-line equation. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, set native vs. manual equation tag mode, install error message for using `\.`. Hand over to mode-specific opening:

```
4376 \def\eql@single@start{%
4377   \eql@display@enter
4378   \eql@tagging@start
4379   \eql@dollar@dollar@begin
4380   \eql@display@adjust
4381   \eql@numbering@init
4382   \eql@stack@save@equations
4383   \eql@numbering@single@init
4384   \ifdefined\eql@single@crerror\else
4385     \let\\\eql@single@cr
4386   \fi
4387   \ifdefined\eql@single@native
4388     \let\eql@single@start@sel\eql@single@start@native
4389     \let\eql@single@end@sel\eql@single@end@native
4390   \else
4391     \let\eql@single@start@sel\eql@single@start@print
4392     \let\eql@single@end@sel\eql@single@end@print
4393   \fi
4394   \eql@single@start@sel
4395 }
```

`\eql@single@end` Closing code for single-line equation. Apply punctuation for the block, perform mode-specific ending, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```
4396 \def\eql@single@end{%
4397   \eql@punct@apply@block
4398   \eql@hook@eqout
4399   \eql@single@end@sel
4400   \eql@stack@restore
4401   \eql@dollar@dollar@end
4402   \eql@tagging@end
4403   \eql@display@leave
4404 }
```

`\eql@single@main` Combined opening, body and closing for pre-scanned body: **TODO:** is `\expandafter` needed? relic?

```
4405 \def\eql@single@main{%
4406   \expandafter\eql@single@start
4407   \eql@scan@body
4408   \eql@single@end
4409 }
```

`\eql@mode@single` Configure equations macros to single-line mode:

```

4410 \def\eql@mode@single{%
4411   \ifdefined\eql@single@doscan
4412     \let\eql@equations@main\eql@single@main
4413     \let\eql@equations@end\@empty
4414   \else
4415     \let\eql@equations@main\@undefined
4416     \let\eql@equations@end\eql@single@end
4417   \fi
4418 }
```

## O.4 Multi-Line Main

`\eql@multi@lines` (*bool*) Switch register for lines vs. columns mode:

```

4419 \let\eql@multi@mode@lines\eql@false
```

`\eql@multi@main` Main routine for multi-line modes. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, initialise macros for use within equations: **TODO:** shove depends on lines vs columns

```

4420 \def\eql@multi@main{%
4421   \eql@display@enter
4422   \eql@tagging@start
4423   \eql@dollar@dollar@begin
4424   \eql@display@adjust
4425   \eql@numbering@init
4426   \eql@stack@save@equations
4427   \ifdefined\eql@transpose@active
4428     \ifdefined\eql@multi@mode@lines\else
4429       \eql@transpose
4430     \fi
4431   \fi
4432   \ifdefined\eql@numbering@subeq@use
4433     \eql@numbering@subeq@init
4434   \fi
4435   \eql@display@init
4436   \let\intertext\eql@intertext
4437   \let\endintertext\endeql@intertext
4438   \eql@shape@align@enable
```

Now measure the given multi-line equations body:

```

4439   \ifdefined\eql@multi@mode@lines
4440     \eql@lines@measure
4441   \else
4442     \ifdefined\eql@ampproof@active
4443       \eql@ampproof
4444     \fi
4445     \eql@columns@measure
4446   \fi
```

If only a single equation number is used for subequation numbering, revert to normal equation numbering. If only a single column is used in columns mode, may fallback to lines mode. Switching from columns to lines mode, the width can be incorrect, expect only minor discrepancies, but for accurateness, should call `\eql@lines@measure`:

```

4447 \ifdefined\eqL@numbering@subeq@use
4448   \eqL@numbering@subeq@test
4449 \fi
4450 \ifdefined\eqL@multi@mode@lines\else
4451   \ifdefined\eqL@multi@linesfallback
4452     \ifnum\eqL@totalcolumns@=\@ne
4453       \let\eqL@multi@mode@lines\eqL@true
4454       \ifx\eqL@multi@linesfallback\z@\else
4455         \eqL@lines@measure
4456       \fi
4457     \fi
4458   \fi
4459 \fi

```

Adjust the multi-line equations body:

```

4460 \ifdefined\eqL@multi@mode@lines
4461   \eqL@lines@adjust
4462 \else
4463   \eqL@columns@adjust
4464 \fi

```

Now print the multi-line equations body:

```

4465 \eqL@display@print
4466 \eqL@numbering@print@init
4467 \ifdefined\eqL@multi@mode@lines
4468   \eqL@lines@print
4469 \else
4470   \eqL@columns@print
4471 \fi
4472 \eqL@display@close

```

Close numbering, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```

4473 \ifdefined\eqL@numbering@subeq@use
4474   \eqL@numbering@subeq@close
4475 \fi
4476 \eqL@stack@restore
4477 \eqL@dollar@end
4478 \eqL@tagging@end
4479 \eqL@display@leave
4480 }

```

`\eqL@mode@columns` Configure equations macros to one of the two multi-line modes:  
`\eqL@mode@lines`

```

4481 \def\eqL@mode@columns{%
4482   \let\eqL@equations@main\eqL@multi@main
4483   \let\eqL@equations@end\@empty
4484   \let\eqL@multi@mode@lines\eqL@false
4485 }
4486 \def\eqL@mode@lines{%
4487   \let\eqL@equations@main\eqL@multi@main
4488   \let\eqL@equations@end\@empty
4489   \let\eqL@multi@mode@lines\eqL@true
4490 }

```



## O.5 Equations Environment

We now declare the main environment and its symbolic versions.

### Environment.

**equations** (*env.*) Declare the main equations environment. If already in math mode, fail and cancel the environment body. Otherwise scan for optional arguments and pass on to `\eqlequations@start`:

```

4491 \newenvironment{equations}{%
4492 (dev)\eql@dev@enterenv
4493   \ifmmode
4494     \eqLError@mathmode{\string\begin{\@currenvir}}%
4495     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4496   \else
4497     \expandafter\eql@ampprotect\expandafter\eqlequations@testall
4498     \expandafter\eqlequations@start
4499   \fi
4500 }{%
4501   \eqlequations@end
4502   \ignorespacesafterend
4503 (dev)\eql@dev@leaveenv
4504 }
4505 \eql@markline@amsthm@register{equations}

```

`\eqlequations@start` The macro `\eqlequations@start` first processes the arguments. Depending on the chosen mode of operation, scan the environment body passing on to `\eqlequations@main` or process a single-line equation via `\eql@single@start`:

```

4506 \def\eqlequations@start{%
4507   \eqlequations@processopt
4508   \ifdefined\eqlequations@main
4509     \expandafter\eql@scan@env\expandafter\eqlequations@main
4510   \else
4511     \expandafter\eql@single@start
4512   \fi
4513 }

```

### Square Brackets.

**equations@sqr** (*env.*) Define a pseudo-environment `equations@sqr` such that `\@currenvir` may point to it when needed:

```

4514 \newenvironment{equations@sqr}{}{}
4515 \eql@markline@amsthm@register{equations@sqr}

```

`\eqlequations@sqr@open` Definition for ‘`\[`’. If already in math mode, ignore the enclosed contents. Otherwise add the default arguments `\eqlequations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eqlequations@sqr@start`:

```

4516 \protected\def\eqlequations@sqr@open{%
4517   \ifmmode
4518     \eqLError@mathmode{\string\[...\string\]}%
4519     \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
4520   \else
4521     (dev)\eql@dev@enter{\[...\string\]}%

```

```

4522 \expandafter\eqnaddopt\expandafter{\eqlequations@sqr@opt}%
4523 \begin{equations@sqr}%
4524 \let\]\eqlequations@sqr@close
4525 \expandafter\eq\amp\protect\expandafter\eqlequations@testall
4526 \expandafter\eqlequations@sqr@start
4527 \fi
4528 }

```

`@equations@sqr@start` Process arguments. Depending on mode of operation, scan and process enclosed contents via `\eqlequations@main` or pass on to `\eqle@single@start`:

```

4529 \def\eqlequations@sqr@start{%
4530 \eqlequations@processopt
4531 \ifdefined\eqlequations@main
4532 \expandafter\eq\scan@sqr\expandafter\eqlequations@main
4533 \else
4534 \expandafter\eqle@single@start
4535 \fi
4536 }

```

`@equations@sqr@close` Definition for ‘\’:

```

4537 \protected\def\eqlequations@sqr@close{%
4538 \eqlequations@end
4539 (dev)\eq\dev@leave{\[...\string\]}%
4540 \end{equations@sqr}%
4541 \ignorespaces
4542 }

```

**TODO:** describe

```

\eqle@sqr@open
\eqle@sqr@close
4543 \let\eqle@sqr@open\eqlequations@sqr@open
4544 \protected\def\eqle@sqr@close{%
4545 \eqle@error{'\string\'} may only close '\string\[']%
4546 }

```

## Angle Brackets.

`equations@ang` (*env.*) Define a pseudo-environment `equations@ang`:

```

4547 \newenvironment{equations@ang}{}{}
4548 \newenvironment{equationsbox@ang}{}{}
4549 \eqle@markline@amsthm@register{equations@ang}

```

`\eqle@ang@open` Definition for ‘\<’. Forward to `equationsbox` if in math mode, otherwise to `equations`:

```

4550 \protected\def\eqle@ang@open{%
4551 (dev)\eq\dev@enter{\<...\string\>}%
4552 \ifmmode
4553 \expandafter\eqnaddopt\expandafter{\eqle@box@ang@opt}%
4554 \begin{equationsbox@ang}%
4555 \let\>\eqle@box@ang@close
4556 \expandafter\eq\amp\protect\expandafter\eqle@box@testall
4557 \expandafter\eqle@box@start
4558 \else
4559 \expandafter\eqnaddopt\expandafter{\eqlequations@ang@opt}%
4560 \begin{equations@ang}%

```

```

4561 \let\>\eqlequations@ang@close
4562 \expandafter\eql@ampprotect\expandafter\eqlequations@testall
4563 \expandafter\eqlequations@ang@start
4564 \fi
4565 }

```

`\eql@ang@close` Definition for ‘`\>`’: **TODO:** NOTE: `\protected` acts as `\relax` and starts a row in `\halign`, so we overwrite `\>` when starting.

```

4566 \protected\def\eql@ang@close{%
4567 \eql@error{'\string\>' may only close '\string\<'}%\>
4568 }

```

`@equations@ang@start` Process arguments and start handling the equation:

```

4569 \def\eqlequations@ang@start{%
4570 \eqlequations@processopt
4571 \ifdefined\eqlequations@main
4572 \expandafter\eql@scan@ang\expandafter\eqlequations@main
4573 \else
4574 \expandafter\eql@single@start
4575 \fi
4576 }

```

`@equations@ang@close` **TODO:** describe

```

4577 \def\eqlequations@ang@close{%
4578 \eqlequations@end
4579 \end{equations@ang}%
4580 (dev)\eql@dev@leave{\<...\string\>}%
4581 \ignorespaces
4582 }

```

`\eql@box@ang@close` **TODO:** describe

```

4583 \def\eql@box@ang@close{%
4584 \eql@box@end
4585 \end{equationsbox@ang}%
4586 (dev)\eql@dev@leave{\<...\string\>}%
4587 \ignorespaces
4588 }

```

## P Options

### P.1 Selection Tools

`eql@decide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```

4589 \def\eql@decide@abovebelow#1#2#3#4#5{%
4590 \eql@decide@select{#1}{#2}{#3}{%
4591 {,abovebelow,both,tb}{#4#5},%
4592 {above,top,t}{#4},%
4593 {below,bottom,b}{#5}}

```

`eql@decide@situation` Select a particular vertical spacing situation and store it in the macro #4:

```

4594 \def\eq@decide@situation#1#2#3#4{%
4595   \eq@decide@select{#1}{#2}{#3}{%
4596     {{long}}{\def#4{0}}},%
4597     {{short}}{\def#4{1}}},%
4598     {{cont}}{\def#4{2}}},%
4599     {{par}}{\def#4{3}}},%
4600     {{top}}{\def#4{4}}},%
4601     {{noskip}}{\def#4{5}}},%
4602     {{medskip}}{\def#4{6}}}}

```

## P.2 Options Declarations

We now declare all key-value pairs for options sorted by their category.

**Modes for Equations Box Environment.** Declare horizontal and vertical alignment modes for the boxed equations environment. Also declare spacing of columns:

```

4603 \eq@define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{}%
4604   \eq@mode@stacked}
4605 \eq@define@key{equationsbox}{aligned,align,al,columns,col}[]{}%
4606   \eq@mode@aligned}
4607 \eq@define@key{equationsbox}{top,t}[]{\let\eq@box@box\vtop}
4608 \eq@define@key{equationsbox}{center,c}[]{\let\eq@box@box\vcenter}
4609 \eq@define@key{equationsbox}{bottom,b}[]{\let\eq@box@box\vbox}
4610 \eq@define@key{setup}{boxangopt}[]{}%
4611   \def\eq@box@ang@opt{columns,#1}}

```

**Modes for Equations Environment.** Declare modes and switches for the equations environment:

```

4612 \eq@define@key{equations}{equation,eq,single,1}[]{\eq@mode@single}
4613 \eq@define@key{equations}{gathered,gather,ga,lines,ln}[]{}%
4614   \eq@mode@lines}
4615 \eq@define@key{equations}{aligned,align,al,columns,col}[]{}%
4616   \eq@mode@columns}
4617 \eq@define@key{equations,setup}{transpose}[true]{}%
4618   \eq@decide@select{#3}{#2}{#1}{%
4619     {\eq@decide@false{\let\eq@transpose@active\eq@false}},%
4620     {\noamp,plain,restricted}{\let\eq@transpose@active\eq@true}},%
4621     {\eq@decide@true,amp,cont}{\let\eq@transpose@active=+}}}%
4622 \eq@define@key{equations}{native}[true]{}%
4623   \eq@decide@bool{#3}{#2}{#1}\eq@single@native%
4624   \ifdefined\eq@single@native\let\eq@layoutleft\eq@false\fi}
4625 \eq@define@key{setup}{native}[true]{}%
4626   \eq@decide@bool{#3}{#2}{#1}\eq@single@native}
4627 \eq@define@key{setup}{scanequation}[true]{}%
4628   \eq@decide@bool{#3}{#2}{#1}\eq@single@doscan}
4629 \eq@define@key{setup}{sqropt}[]{}%
4630   \def\eq@equations@sqr@opt{equation,#1}}
4631 \eq@define@key{setup}{angopt}[]{}%
4632   \def\eq@equations@ang@opt{columns,#1}}

```

**Vertical Spacing.** Settings concerning the spacing of lines: **TODO:** set at end of env only!

```

4633 \def\eq@keycat{equations,equationsbox,setup}

```

```

4634 \eqld@define@key\eqld@keycat{spread}{\def\eqld@spread@val{#1}}
4635 \eqld@define@key\eqld@keycat{strut}[true]{\eqld@decide@select{#3}{#2}{#1}{%
4636   {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%
4637   {{cell}{\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\relax}},%
4638   {{tag}{\let\eqld@strut@cell\relax\let\eqld@strut@tag\eqld@strut}},%
4639   {\eqld@decide@true
4640     {\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\eqld@strut}}}}
4641 \eqld@define@key{setup}{strutdepth}{\def\eqld@strut@depth{#1}}

```

Settings concerning page breaks:

```

4642 \eqld@define@key{equations}{prebreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
4643   {{force,4,\eqld@decide@true}{\eqld@displaybreak@pre4}},%
4644   {{high,3}{\eqld@displaybreak@pre3}},%
4645   {{med,medium,2}{\eqld@displaybreak@pre2}},%
4646   {{low,1}{\eqld@displaybreak@pre1}},%
4647   {{0,\eqld@decide@false}{\eqld@displaybreak@pre0}},%
4648   {{default,inherit,-1}{\eqld@displaybreak@pre\m@ne}}}}
4649 \eqld@define@key{equations}{postbreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
4650   {{force,4,\eqld@decide@true}{\eqld@displaybreak@post4}},%
4651   {{high,3}{\eqld@displaybreak@post3}},%
4652   {{med,medium,2}{\eqld@displaybreak@post2}},%
4653   {{low,1}{\eqld@displaybreak@post1}},%
4654   {{0,\eqld@decide@false}{\eqld@displaybreak@post0}},%
4655   {{default,inherit,-1}{\eqld@displaybreak@post\m@ne}}}}
4656 \eqld@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4657   \eqld@decide@select{#3}{#2}{#1}{%
4658     {{full,4}{\eqld@displaybreak@inter4}},%
4659     {{high,3}{\eqld@displaybreak@inter3}},%
4660     {{med,medium,2}{\eqld@displaybreak@inter2}},%
4661     {{low,1}{\eqld@displaybreak@inter1}},%
4662     {{0,\eqld@decide@false}{\eqld@displaybreak@inter\z@}}}}
4663 \eqld@define@key{equations}{prepenalty}{%
4664   \eqld@displaybreak@prepen@\numexpr#1\relax}
4665 \eqld@define@key{equations}{postpenalty}{%
4666   \eqld@displaybreak@postpen@\numexpr#1\relax}
4667 \eqld@define@key{equations,setup}{interpenalty}{%
4668   \interdisplaylinepenalty\numexpr#1\relax}

```

**TODO:** describe

```

4669 \eqld@define@key{control}{vspace}[]{\eqld@vspace@add{#1}}
4670 \eqld@define@key{control}{vspace*}[]{\eqld@vspace@addfixedbefore{#1}}
4671 \eqld@define@key{control}{vspace!}[]{\eqld@vspace@addfixedafter{#1}}
4672 \eqld@define@key{control}{break}[4]{\eqld@displaybreak@level[#{#1}]}
4673 \eqld@define@key{control}{penalty}[]{\eqld@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4674 \eqld@define@key\eqld@keycat{displayheight}[strut]{%
4675   \eqld@decide@select{#3}{#2}{#1}{%
4676     {\eqld@decide@false{\let\eqld@display@height\@undefined}},%
4677     {{strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
4678     {\relax{\def\eqld@display@height{#1}}}}}}
4679 \eqld@define@key\eqld@keycat{displaydepth}[strut]{%
4680   \eqld@decide@select{#3}{#2}{#1}{%
4681     {\eqld@decide@false{\let\eqld@display@depth\@undefined}},%
4682     {{strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
4683     {\relax{\def\eqld@display@depth{#1}}}}}}

```

Override vertical spacing situation: **TODO:** short should just apply to above?! or as far

as short would apply...

```

4684 \eqld@define@key{equations}{noskip}[] {%
4685   \eqld@decide@abovebelow{#3}{#2}{#1}%
4686   {\def\eqld@skip@force@above{5}}%
4687   {\def\eqld@skip@force@below{5}}}
4688 \eqld@define@key{equations}{short}[above] {%
4689   \eqld@decide@abovebelow{#3}{#2}{#1}%
4690   {\def\eqld@skip@force@above{1}}%
4691   {\def\eqld@skip@force@below{1}}}
4692 \eqld@define@key{equations}{long}[] {%
4693   \eqld@decide@abovebelow{#3}{#2}{#1}%
4694   {\def\eqld@skip@force@above{0}}%
4695   {\def\eqld@skip@force@below{0}}}
4696 \eqld@define@key{equations}{medskip}[] {%
4697   \eqld@decide@abovebelow{#3}{#2}{#1}%
4698   {\def\eqld@skip@force@above{6}}%
4699   {\def\eqld@skip@force@below{6}}}
4700 \eqld@define@key{equations}{par}[par] {%
4701   \eqld@decide@select{#3}{#2}{#1}{%
4702     {{default,}{\let\eqld@skip@force@leave\undefined}},%
4703     {{cont,hmode}{\let\eqld@skip@force@leave\z@}},%
4704     {{par,vmode}{\let\eqld@skip@force@leave\@ne
4705       \ifdefined\eqld@skip@force@below\else
4706         \def\eqld@skip@force@below{3}%
4707       \fi}},%
4708     {{top}{\let\eqld@skip@force@leave\tw@
4709       \ifdefined\eqld@skip@force@below\else
4710         \def\eqld@skip@force@below{4}
4711       \fi}}}}

```

Specify vertical spacing explicitly:

```

4712 \eqld@define@key{equations}{skip}{%
4713   \def\eqld@skip@force@above{7}%
4714   \def\eqld@skip@custom@above{#1}%
4715   \let\eqld@skip@force@below\eqld@skip@force@above
4716   \let\eqld@skip@custom@below\eqld@skip@custom@above}
4717 \eqld@define@key{equations}{aboveskip}{%
4718   \def\eqld@skip@force@above{7}%
4719   \def\eqld@skip@custom@above{#1}}
4720 \eqld@define@key{equations}{belowskip}{%
4721   \def\eqld@skip@force@below{7}%
4722   \def\eqld@skip@custom@below{#1}}
4723 \eqld@define@key{equations}{abovespace}{%
4724   \advance\eqld@abovespace@glueexpr#1\relax}
4725 \eqld@define@key{equations}{belowspace}{%
4726   \advance\eqld@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4727 \eqld@define@key{intertext}{skip}{%
4728   \def\eqld@skip@force@above{7}%
4729   \def\eqld@skip@custom@above{#1}%
4730   \let\eqld@skip@force@below\eqld@skip@force@above
4731   \let\eqld@skip@custom@below\eqld@skip@custom@above}
4732 \eqld@define@key{intertext}{aboveskip}{%
4733   \def\eqld@skip@force@below{7}%
4734   \def\eqld@skip@custom@below{#1}}
4735 \eqld@define@key{intertext}{belowskip}{%

```

```

4736 \def\eql@skip@force@above{7}%
4737 \def\eql@skip@custom@above{#1}%
4738 \eql@define@key{intertext}{noskip}[]{%
4739 \eql@decide@abovebelow{#3}{#2}{#1}%
4740 {\def\eql@skip@force@below{5}}%
4741 {\def\eql@skip@force@above{5}}}%
4742 \eql@define@key{intertext}{short}[]{%
4743 \eql@decide@abovebelow{#3}{#2}{#1}%
4744 {\def\eql@skip@force@below{1}}%
4745 {\def\eql@skip@force@above{1}}}%
4746 \eql@define@key{intertext}{long}[]{%
4747 \eql@decide@abovebelow{#3}{#2}{#1}%
4748 {\def\eql@skip@force@below{0}}%
4749 {\def\eql@skip@force@above{0}}}%
4750 \eql@define@key{intertext}{medskip}[]{%
4751 \eql@decide@abovebelow{#3}{#2}{#1}%
4752 {\def\eql@skip@force@below{6}}%
4753 {\def\eql@skip@force@above{6}}}%

```

Configure general vertical spacing behaviour for various situations:

```

4754 \eql@define@key{setup}{skip,longskip}{%
4755 \abovedisplayskip\glueexpr#1\relax
4756 \belowdisplayskip\abovedisplayskip
4757 \def\eql@skip@long@above{#1}%
4758 \let\eql@skip@long@below\eql@skip@long@above}
4759 \eql@define@key{setup}{aboveskip,abovelongskip}{%
4760 \abovedisplayskip\glueexpr#1\relax
4761 \def\eql@skip@long@above{#1}}
4762 \eql@define@key{setup}{belowskip,belowlongskip}{%
4763 \belowdisplayskip\glueexpr#1\relax
4764 \def\eql@skip@long@below{#1}}
4765 \eql@define@key{setup}{aboveshortskip}{%
4766 \abovedisplayshortskip\glueexpr#1\relax
4767 \def\eql@skip@short@above{#1}}
4768 \eql@define@key{setup}{belowshortskip}{%
4769 \belowdisplayshortskip\glueexpr#1\relax
4770 \def\eql@skip@short@below{#1}}
4771 \eql@define@key{setup}{tagskip}{%
4772 \def\eql@skip@tag@above{#1}%
4773 \let\eql@skip@tag@below\eql@skip@tag@above}
4774 \eql@define@key{setup}{abovetagskip}{%
4775 \def\eql@skip@tag@above{#1}}
4776 \eql@define@key{setup}{belowtagskip}{%
4777 \def\eql@skip@tag@below{#1}}
4778 \eql@define@key{setup}{medskip}{%
4779 \def\eql@skip@med@above{#1}%
4780 \let\eql@skip@med@below\eql@skip@med@above}
4781 \eql@define@key{setup}{abovemedskip}{%
4782 \def\eql@skip@med@above{#1}}
4783 \eql@define@key{setup}{belowmedskip}{%
4784 \def\eql@skip@med@below{#1}}
4785 \eql@define@key{setup}{abovetopskip}{%
4786 \def\eql@skip@top@above{#1}}
4787 \eql@define@key{setup}{belowtopskip}{%
4788 \def\eql@skip@top@below{#1}}
4789 \eql@define@key{setup}{aboveparskip}{%
4790 \def\eql@skip@par@above{#1}}
4791 \eql@define@key{setup}{belowparskip}{%

```

```

4792 \def\eql@skip@par@below{#1}}
4793 \eql@define@key{setup}{abovecontskip}{%
4794 \eql@decide@select{#3}{#2}{#1}{%
4795   {{hide}}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%
4796   {\relax{\def\eql@skip@cont@above{#1}}}}
4797 \eql@define@key{setup}{belowcontskip}{%
4798 \def\eql@skip@cont@below{#1}}
4799 \eql@define@key{setup}{shortmode}{%
4800 \eql@decide@select{#3}{#2}{#1}{%
4801   {{off,never,no}}{\def\eql@skip@mode@short{0}}},%
4802   {{above,neverbelow,notbelow,belowoff}}{\def\eql@skip@mode@short{1}}},%
4803   {{belowone,belowsingle}}{\def\eql@skip@mode@short{2}}},%
4804   {{belowall,always,on}}{\def\eql@skip@mode@short{3}}}}
4805 \eql@define@key{setup}{abovecontmode}{%
4806 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}
4807 \eql@define@key{setup}{belowcontmode}{%
4808 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
4809 \eql@define@key{setup}{aboveparmode}{%
4810 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
4811 \eql@define@key{setup}{belowparmode}{%
4812 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
4813 \eql@define@key{setup}{abovetopmode}{%
4814 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
4815 \eql@define@key{setup}{belowtopmode}{%
4816 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

**Labels and Tag Declaration.** Specify label and tag for equations and subequations:

```

4817 \def\eql@keycat{equations,subequations}
4818 \eql@define@key\eql@keycat{label}{\eql@tags@addblock@label{#1}}
4819 \eql@define@key\eql@keycat{labelname}{\eql@tags@addblock@name{#1}}
4820 \eql@define@key\eql@keycat{tag}{\eql@tags@addblock@tag{#1}}
4821 \eql@define@key\eql@keycat{tag*}{%
4822 \eql@tags@addblock@tagform@off\eql@tags@addblock@tag{#1}}
4823 \eql@define@key\eql@keycat{taglabel}{\eql@tags@addblock@ref{#1}}

```

**TODO:** describe

```

4824 \eql@define@key{control}{label}{\eql@tags@add@label{#1}}
4825 \eql@define@key{control}{labelname}{\eql@tags@add@name{#1}}
4826 \eql@define@key{control}{tag}{\eql@tags@add@tag{#1}}
4827 \eql@define@key{control}{tag*}{\eql@tags@add@tagform@off\eql@tags@add@tag{#1}}
4828 \eql@define@key{control}{taglabel}{\eql@tags@add@ref{#1}}
4829 \eql@define@key{control}{shifttag}{\eql@tags@add@raiseshift{#1}}
4830 \eql@define@key{control}{smashtag}{\eql@tags@add@raisesmash{#1}}
4831 \eql@define@key{control}{pushtag}[]{\eql@tags@add@forceraise}

```

**TODO:** describe

```

4832 \eql@define@key{setup}{labelname}{\protected@edef\eql@tags@name@generic{#1}}
4833 \eql@define@key{setup}{autolabel}[true]{%
4834 \eql@decide@bool{#3}{#2}{#1}\eql@tags@autolabel}
4835 \eql@define@key{setup}{autotag}[true]{%
4836 \eql@decide@bool{#3}{#2}{#1}\eql@tags@autotag}

```

**Tag Spacing.** Configure horizontal spacing for equation tags:

```

4837 \def\eql@keycat{equations,setup}
4838 \eql@define@key\eql@keycat{tagmargin}[auto]{%

```



```

4839 \eqld@decide@select{#3}{#2}{#1}{%
4840   {{\auto,\eqld@decide@false}{\let\eqld@tagmargin@val\undefined}},%
4841   {\relax{\def\eqld@tagmargin@val{#1}}}}}%
4842 \eqld@define@key\eqld@keycat{tagmargin*}{%
4843   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
4844 \eqld@define@key\eqld@keycat{tagmarginratio}{%
4845   \eqld@tagmargin@ratio@{\dimexpr#1pt\relax}}
4846 \eqld@define@key\eqld@keycat{tagmarginthreshold}{%
4847   \def\eqld@tagmargin@threshold{#1}}
4848 \eqld@define@key\eqld@keycat{mintagsep}{\def\eqld@tagsepmin@val{#1}}
4849 \eqld@define@key\eqld@keycat{mintagwidth}{%
4850   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
4851 \eqld@define@key\eqld@keycat{mintagwidth*}{\settowidth\eqld@tagwidthmin@{#1}}
4852 \eqld@define@key\eqld@keycat{tagsnap}{%
4853   \eqld@decide@select{#3}{#2}{#1}{%
4854     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
4855     {\relax{\def\eqld@tagpos@snap{#1}}}}}%

```

**Tag Layout.** Configure methods to declare equation tag layout:

```

4856 \def\eqld@keycat{equations,setup}
4857 \eqld@define@key\eqld@keycat{tagbox,taglayout}{%
4858   \eqld@tags@taglayout@set{#1}}
4859 \eqld@define@key\eqld@keycat{tagbox*,taglayout*}{%
4860   \eqld@tags@taglayout@set@direct{#1}}
4861 \eqld@define@key\eqld@keycat{tagform}{%
4862   \eqld@tags@tagform@set{#1}}
4863 \eqld@define@key\eqld@keycat{tagform*}{%
4864   \eqld@tags@tagform@set@direct{#1}}
4865 \eqld@define@key\eqld@keycat{subeqtemplate}{%
4866   \def\eqld@subequations@template####1####2{#1}%
4867   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}}

4868 \eqld@define@key{control}{tagbox,taglayout}{%
4869   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
4870 \eqld@define@key{control}{tagbox*,taglayout*}{%
4871   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
4872 \eqld@define@key{control}{tagform}{%
4873   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
4874 \eqld@define@key{control}{tagform*}{[####1]}{%
4875   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

**Equation Numbering.** Configure equation numbering schemes:

```

4876 \def\eqld@keycat{equations,setup}
4877 \eqld@define@key\eqld@keycat{numberline,number,num,numline,n}[all]{%
4878   \eqld@decide@select{#3}{#2}{#1}{%
4879     {{\eqld@decide@false,0,*}{\let\eqld@numbering@active\eqld@false}},%
4880     {{\eqld@decide@true,!}{\let\eqld@numbering@active\eqld@true}},%
4881     {{none,n,-}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
4882       \let\eqld@numbering@active\eqld@false}},%
4883     {{single,1}{\let\eqld@numbering@mode\eqld@numbering@mode@single
4884       \let\eqld@numbering@active\eqld@true}},%
4885     {{multi,0}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
4886       \let\eqld@numbering@active\eqld@true}},%
4887     {\relax{\eqld@numbering@set{#1}}}}}%
4888 \eqld@define@key\eqld@keycat{nonumber,nn,*}[]{%
4889   \let\eqld@numbering@active\eqld@false}

```

```

4890 \eqld@define@key\eqld@keycat{donumber,dn,!}[]{%
4891   \let\eqld@numbering@active\eqld@true}
4892 \eqld@define@key\eqld@keycat{tagsleft,leqno}[]{\let\eqld@tagsleft\eqld@true}
4893 \eqld@define@key\eqld@keycat{tagsright,reqno}[]{\let\eqld@tagsleft\eqld@false}
4894 \eqld@define@key\eqld@keycat{tags,eqno}{%
4895   \eqld@decide@select{#3}{#2}{#1}{%
4896     {{right,r}}{\let\eqld@tagsleft\eqld@false}},%
4897     {{left,l}}{\let\eqld@tagsleft\eqld@true}}}}
4898 \eqld@define@key\eqld@keycat{evadetag,avoidtag}[true]{%
4899   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}
4900 \eqld@define@key\eqld@keycat{tagbetween}[true]{%
4901   \eqld@decide@bool{#3}{#2}{#1}\eqld@tagpos@doconvert}

```

**TODO:** describe

```

4902 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
4903 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
4904 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
4905 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

**Horizontal Layout.** Configure horizontal alignment mode and margin for left alignment:

```

4906 \def\eqld@keycat{equations,setup}
4907 \eqld@define@key\eqld@keycat{layout}{\eqld@decide@select{#3}{#2}{#1}{%
4908   {{center,c}}{\let\eqld@layoutleft\eqld@false}},%
4909   {{left,l}}{\let\eqld@layoutleft\eqld@true}}}}
4910 \eqld@define@key\eqld@keycat{center}[]{\let\eqld@layoutleft\eqld@false}
4911 \eqld@define@key\eqld@keycat{flushleft,left}[]{\let\eqld@layoutleft\eqld@true}
4912 \eqld@define@key\eqld@keycat{leftmargin}{\def\eqld@layoutleftmargin{#1}}
4913 \eqld@define@key\eqld@keycat{leftmargin*}{%
4914   \settowidth\dimen@{#1}\edef\eqld@layoutleftmargin{\the\dimen@}}
4915 \eqld@define@key\eqld@keycat{minleftmargin}{%
4916   \def\eqld@layoutleftmarginmin{#1}}
4917 \eqld@define@key\eqld@keycat{maxleftmargin}{%
4918   \eqld@decide@select{#3}{#2}{#1}{%
4919     {\eqld@decide@false{\def\eqld@layoutleftmarginmax{.5\maxdimen}}},%
4920     {\relax{\def\eqld@layoutleftmarginmax{#1}}}}}
4921 \def\eqld@keycat{equations,equationsbox}
4922 \eqld@define@key\eqld@keycat{margin}{%
4923   \def\eqld@display@marginleft{#1}\def\eqld@display@marginright{#1}}
4924 \eqld@define@key\eqld@keycat{marginleft}{\def\eqld@display@marginleft{#1}}
4925 \eqld@define@key\eqld@keycat{marginright}{\def\eqld@display@marginright{#1}}
4926 \eqld@define@key{equations}{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

**Horizontal Spacing and Columns.** Configure column spacing and compression threshold:

```

4927 \def\eqld@keycat{equations,setup}
4928 \eqld@define@key\eqld@keycat{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
4929   {{max,full,4}}{\eqld@alignbadness@inf@bad}},%
4930   {{high,3}}{\eqld@alignbadness@54@relax}},%
4931   {{med,medium,2}}{\eqld@alignbadness@18@relax}},%
4932   {{low,1}}{\eqld@alignbadness@6@relax}},%
4933   {{0,\eqld@decide@false}}{\eqld@alignbadness@z@}}}
4934 \eqld@define@key\eqld@keycat{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
4935   {{max,full,4}}{\eqld@tagbadness@inf@bad}},%

```

```

4936    {{high,3}}{\eq@tagbadness@54\relax}},%
4937    {{med,medium,2}}{\eq@tagbadness@18\relax}},%
4938    {{low,1}}{\eq@tagbadness@6\relax}},%
4939    {{0,\eq@decide@false}}{\eq@tagbadness@z@{}}}}
4940 \eq@define@key\eq@keycat{alignbadness}{\eq@alignbadness@numexpr#1\relax}
4941 \eq@define@key\eq@keycat{tagbadness}{\eq@tagbadness@numexpr#1\relax}
4942 \eq@define@key\eq@keycat{mincolsep}{\eq@decide@select{#3}{#2}{#1}{%
4943     {{0,\eq@decide@false}}{\def\eq@colsepmin@val{0pt}}}},%
4944     {\relax{\def\eq@colsepmin@val{#1}}}}}}
4945 \eq@define@key\eq@keycat{maxcolsep}{\eq@decide@select{#3}{#2}{#1}{%
4946     {\eq@decide@false{\def\eq@colsepmax@val{.5\maxdimen}}}},%
4947     {\relax{\def\eq@colsepmax@val{#1}}}}}}
4948 \eq@define@key\eq@keycat{fulllength}[true]{%
4949     \eq@decide@bool{#3}{#2}{#1}\eq@columns@fulllength}

4950 \eq@define@key{equationsbox,setup}{colsep}{\eq@decide@select{#3}{#2}{#1}{%
4951     {{0,\eq@decide@false}}{\def\eq@box@colsep{0pt}}}},%
4952     {\relax{\def\eq@box@colsep{#1}}}}}}
4953 \eq@define@key{equations}{colsep}{\eq@decide@select{#3}{#2}{#1}{%
4954     {{0,\eq@decide@false}}{\def\eq@box@colsep{0pt}}}},%
4955     {\relax{\def\eq@box@colsep{#1}}}}}}
4956 \let\eq@colsepmin@val\eq@box@colsep
4957 \let\eq@colsepmax@val\eq@box@colsep

```

**Horizontal Shape.** Configure horizontal alignment schemes:

```

4958 \def\eq@keycat{equations,equationsbox,setup}
4959 \eq@define@key\eq@keycat{shape}[default]{\eq@shape@set{#1}}
4960 \eq@define@key\eq@keycat{padding,pad}[indent]{%
4961     \eq@decide@select{#3}{#2}{#1}{%
4962         {{max}}{\let\eq@paddingleft@val\@undefined}},%
4963         {{indent}}{\def\eq@paddingleft@val{\eq@indent@val}}}},%
4964         {{0,\eq@decide@false}}{\def\eq@paddingleft@val{0pt}}}},%
4965         {\relax{\def\eq@paddingleft@val{#1}}}}}}
4966 \let\eq@paddingright@val\eq@paddingleft@val
4967 \eq@define@key\eq@keycat{padleft}[indent]{%
4968     \eq@decide@select{#3}{#2}{#1}{%
4969         {{max}}{\let\eq@paddingleft@val\@undefined}},%
4970         {{indent}}{\def\eq@paddingleft@val{\eq@indent@val}}}},%
4971         {{0,\eq@decide@false}}{\def\eq@paddingleft@val{0pt}}}},%
4972         {\relax{\def\eq@paddingleft@val{#1}}}}}}
4973 \eq@define@key\eq@keycat{padright}[indent]{%
4974     \eq@decide@select{#3}{#2}{#1}{%
4975         {{max}}{\let\eq@paddingright@val\@undefined}},%
4976         {{indent}}{\def\eq@paddingright@val{\eq@indent@val}}}},%
4977         {{0,\eq@decide@false}}{\def\eq@paddingright@val{0pt}}}},%
4978         {\relax{\def\eq@paddingright@val{#1}}}}}}
4979 \eq@define@key\eq@keycat{indent}[2em]{%
4980     \def\eq@indent@val{#1}}

```

**TODO:** describe

```

4981 \eq@define@key{control}{align}[]{%
4982     \eq@decide@select{#3}{#2}{#1}{%
4983         {{l,left}}{\global\eq@append\eq@cell@container{\eq@shape@pos@z@{}}},%
4984         {{c,center}}{\global\eq@append\eq@cell@container{\eq@shape@pos@one@{}}},%
4985         {{r,right}}{\global\eq@append\eq@cell@container{\eq@shape@pos@tw@{}}}}}}
4986 \eq@define@key{control}{shift,shifto}[]{%
4987     \eq@decide@select{#3}{#2}{#1}{%

```

```

4988    {\*,indent}\@eq\shape@alignamount@set{\@eq\indent0}}},%
4989    {\!,outdent}\@eq\shape@alignamount@set{-\@eq\indent0}}},%
4990    {\relax\@eq\shape@alignamount@set{#1}}}}
4991 \eq\define@key{control}{shift*,shiftby}[]{\@eq\shape@alignamount@add{#1}}

```

**Math Classes at Alignment.** Configure math classes at alignment marker:

```

4992 \def\eq\keycat{equations,equationsbox,setup}
4993 \eq\define@key\eq\keycat{classout}{\eq\class@innerleft@set{#1}}
4994 \eq\define@key\eq\keycat{classin}{\eq\class@innerright@set{#1}}
4995 \eq\define@key\eq\keycat{classlead,classin*}{\eq\class@innerlead@set{#1}}
4996 \eq\define@key\eq\keycat{ampeq}[]{\eq\class@ampeq}
4997 \eq\define@key\eq\keycat{eqamp}[]{\eq\class@eqamp}
4998 \eq\define@key\eq\keycat{class}{\eq\decide@select{#3}{#2}{#1}{%
4999   {\ampeq,amprel,eafter,beforerel}\eq\class@ampeq},%
5000   {\eqamp,relamp,eqbefore,afterrel}\eq\class@eqamp}}

```

**Punctuation.** Configure punctuation defaults: **TODO:** describe

```

5001 \def\eq\punct@all#1#2#3#4#5\eq\punct@end{%
5002   \def\eq\tmp{#4}\def\eq\tmpa{1}%
5003   \ifx\eq\tmp\eq\tmpa
5004     \ifnum#5=1111\relax
5005       \eq\punct@set\eq\punct@col{#1}%
5006       \eq\punct@set\eq\punct@line{#2}%
5007       \eq\punct@set\eq\punct@main{#3}%
5008     \else\ifnum#5=111\relax
5009       \eq\punct@set\eq\punct@line{#1}%
5010       \eq\punct@set\eq\punct@main{#2}%
5011     \else\ifnum#5=11\relax
5012       \eq\punct@set\eq\punct@main{#1}%
5013     \else
5014       \let\eq\punct@col\@empty
5015       \let\eq\punct@line\@empty
5016       \let\eq\punct@main\@empty
5017     \fi\fi\fi
5018   \else
5019     \eq\error{Too many arguments to punctall}%
5020   \fi
5021 }

```

**TODO:** describe

```

5022 \def\eq\keycat{equations,equationsbox,setup}
5023 \eq\define@key\eq\keycat{punctsep}{\,}{\def\eq\punct@sep{#1}}
5024 \eq\define@key\eq\keycat{punct}[]{\eq\punct@set\eq\punct@main{#1}}
5025 \eq\define@key\eq\keycat{punct*}[]{\let\eq\punct@main\relax}
5026 \eq\define@key\eq\keycat{punctline}[]{\eq\punct@set\eq\punct@line{#1}}
5027 \eq\define@key\eq\keycat{punctline*}[]{\let\eq\punct@line\relax}
5028 \eq\define@key\eq\keycat{punctcol}[]{\eq\punct@set\eq\punct@col{#1}}
5029 \eq\define@key\eq\keycat{punctcol*}[]{\let\eq\punct@col\relax}
5030 \eq\define@key\eq\keycat{punctall}[,]{\eq\punct@all#111111\eq\punct@end}

5031 \eq\define@key{control}{punctsep}{\,}{\def\eq\punct@sep{#1}}
5032 \eq\define@key{control}{punct}[]{\eq\punct@set\eq\punct@block{#1}%
5033   \eq\punct@set\eq\punct@line{#1}\eq\punct@set\eq\punct@col{#1}}
5034 \eq\define@key{control}{punct*}[]{\let\eq\punct@block\relax}
5035 \eq\define@key{control}{punctapply}[]{\eq\punct@apply@block}

```

**Frames.** **TODO:** describe

```
5036 \eqld@define@key{equationsbox}{frame}[\fbox]{%
5037   \def\eql@box@frame{#1}%
5038   \ifx\eql@box@frame\empty\let\eql@box@frame\@firstofone\fi}
5039 \eqld@define@key{equationsbox}{wrap}[]{\eql@box@wrap#1}
```

**TODO:** describe

```
5040 \eqld@define@key{control}{framecell}[\fbox]{%
5041   \global\eqld@append\eqld@cell@container{\def\eqld@frame@cmd{#1}}
5042 \eqld@define@key{control}{frametag}[\fbox]{%
5043   \global\eqld@append\eqld@tags@container{\def\eqld@tags@frame@cmd{#1}}}
```

**Alternative Content Description.** Alternative content description for accessibility or documentation purposes: **TODO:** implement in PDF tagging

```
5044 \eqld@define@key{equations,equationsbox}{alt}{}
```

**Injections.**

```
5045 \eqld@define@key{control}{inject}{%
5046   \global\eqld@append\eqld@interline@container{%
5047     \eqld@append\eqld@display@injectbefore{#1}}
5048 \eqld@define@key{control}{inject*}{%
5049   \global\eqld@append\eqld@interline@container{%
5050     \eqld@append\eqld@display@injectafter{#1}}
5051 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
5052 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
5053 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
5054 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}
```

**TODO:** describe

```
5055 \eqld@define@key{markline}{pos}{%
5056   \eqld@decide@select{#3}{#2}{#1}{%
5057     {{below,push}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5058     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5059     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}%
5060 \eqld@define@key{markline}{below,push}[]{%
5061   \let\eqld@markline@pos\eqld@markline@pos@below}
5062 \eqld@define@key{markline}{baseline}[]{%
5063   \let\eqld@markline@pos\eqld@markline@pos@baseline}
5064 \eqld@define@key{markline}{bottom}[]{%
5065   \let\eqld@markline@pos\eqld@markline@pos@bottom}
5066 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
5067 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}
5068 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
5069 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}
5070 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
5071 \eqld@define@key{setup}{markpos}{%
5072   \eqld@decide@select{#3}{#2}{#1}{%
5073     {{below}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5074     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5075     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}%}
```

**Global Switches.** Set global switches:

```
5076 \let\eqld@multi@linesfallback\eqld@false
```

```

5077 \let\eql@scan@par\eql@false
5078 \let\eql@single@crerror\eql@false
5079 \let\eql@ampproof@active\eql@false

5080 \eql@define@key{equations,setup}{linesfallback}[true]{%
5081   \eql@decide@select{#3}{#2}{#1}{%
5082     {\eql@decide@false{\let\eql@multi@linesfallback\eql@false}},%
5083     {{reuse,lean}{\let\eql@multi@linesfallback\z@}},%
5084     {{measure,full,\eql@decide@true}{\let\eql@multi@linesfallback\eql@true}}}}
5085 \eql@define@key{setup}{ampproof}[true]{%
5086   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
5087 \eql@define@key{setup}{crerror}[true]{%
5088   \eql@decide@bool{#3}{#2}{#1}\eql@single@crerror}
5089 \eql@define@key{setup}{modifierwarning}[true]{%
5090   \eql@decide@select{#3}{#2}{#1}{%
5091     {\eql@decide@false}{\let\eql@parseopt@warn\@empty}},%
5092     {\eql@decide@true}{\let\eql@parseopt@warn\eql@warn@parseopt}},%
5093     {{verbose,+}{\let\eql@parseopt@warn\eql@warn@parseopt@verbose}}}}
5094 \let\eql@parseopt@warn\eql@warn@parseopt
5095 \eql@define@key{equations,setup}{rescan}[true]{%
5096   \eql@decide@if{#3}{#2}{#1}%
5097   {\let\eql@scan@body\eql@scan@body@rescan}%
5098   {\let\eql@scan@body\eql@scan@body@dump}}
5099 \eql@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5100   \eql@decide@bool{#3}{#2}{#1}\eql@scan@par}
5101 \eql@define@key{setup}{defaults}{%
5102   \eql@decide@select{#3}{#2}{#1}{%
5103     {{classic}{\eql@defaults@classic}},%
5104     {{eqnlines}{\eql@defaults@eqnlines}}}}

```

**Package Options.** Declare choices available at loading of package only: **TODO:** adjust

```

5105 \let\eql@provide@opt@env\tw@
5106 \let\eql@provide@opt@amsmathends\eql@true
5107 \let\eql@provide@opt@backup\eql@false
5108 \let\eql@provide@opt@ang\eql@true
5109 \let\eql@provide@opt@eqref\eql@true

5110 \eql@define@key{setup}{amsmathends}[true]{%
5111   \eql@error@packageoption{#2}%
5112   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
5113 \eql@define@key{setup}{backup}[true]{%
5114   \eql@error@packageoption{#2}%
5115   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
5116 \eql@define@key{setup}{env}[equation]{%
5117   \eql@error@packageoption{#2}%
5118   \eql@decide@select{#3}{#2}{#1}{%
5119     {{none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5120     {{equation,latex}{\let\eql@provide@opt@env\@ne}},%
5121     {{amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}}
5122 \eql@define@key{setup}{ang}[true]{%
5123   \eql@error@packageoption{#2}%
5124   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5125 \eql@define@key{setup}{eqref}[true]{%
5126   \eql@error@packageoption{#2}%
5127   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Shortcut Options. **TODO:** describe

```

5128 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5129 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5130 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5131 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5132 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5133 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5134 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5135 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5136 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
5137 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
5138 \def\eql@parseopt@punctoff#1{\eqnaddopt{punct={}}\eql@parseopt@peek}
5139 \def\eql@parseopt@punctall#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}

```

### P.3 Parameter Presets

The package offers two parameter presets which lead to somewhat different layout. Instead of setting the internal parameters directly, we expose them as public settings so that they are easier to read and such that individual settings can be used to compose own layouts.

`\eql@defaults@classic` The preset `classic` aims to reproduce the  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

5140 \def\eql@defaults@classic{%
5141   \eqnlineset{numberline=all}%
5142   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5143   \eqnlineset{maxcolsep=off}%
5144   \eqnlineset{spread={\jot}}%
5145   \eqnlineset{tagmargin}%
5146   \eqnlineset{tagmarginratio=1}%
5147   \eqnlineset{tagmarginthreshold=0.5}%
5148   \eqnlineset{leftmargin={\leftmargini}}%
5149   \eqnlineset{padding=max}%
5150   \eqnlineset{evadetag=off}%
5151   \eqnlineset{displayheight=off}%
5152   \eqnlineset{displaydepth=off}%
5153   \eqnlineset{shortmode=belowsingle}%
5154   \eqnlineset{abovecontmode=short}%
5155   \eqnlineset{belowcontmode=short}%
5156   \eqnlineset{aboveparmode=long}%
5157   \eqnlineset{belowparmode=long}%
5158   \eqnlineset{abovetopmode=long}%
5159   \eqnlineset{belowtopmode=long}%
5160   \eqnlineset{abovelongskip={\abovedisplayskip}}%
5161   \eqnlineset{belowlongskip={\belowdisplayskip}}%
5162   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5163   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5164   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5165   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5166   \eqnlineset{abovecontskip=0pt}%
5167   \eqnlineset{belowcontskip=0pt}%
5168   \eqnlineset{aboveparskip=0pt}%
5169   \eqnlineset{belowparskip=0pt}%
5170   \eqnlineset{abovetopskip=0pt}%
5171   \eqnlineset{belowtopskip=0pt}%
5172   \eqnlineset{abovetagskip=0pt}%
5173   \eqnlineset{belowtagskip=0pt}%

```



```

5174 \eqnlineset{crerror=false}%
5175 \eqnlineset{linesfallback=false}%
5176 }

```

values based on 10pt vs 12pt

`q1@defaults@eqnlines` The (default) preset `eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the `classic` spacing for a 12pt computer modern font such that 10pt fonts will lead to slightly reduced spacing. Apart from that, the `eqnlines` setting makes some deliberate layout choices that deviate significantly from `classic` (maximum column separation, no shortening below equations):

```

5177 \def\eq1@defaults@eqnlines{%
5178   \eqnlineset{numberline=all}%
5179   \eqnlineset{mintagsep=.5em}%
5180   \eqnlineset{maxcolsep=2em}%
5181   \eqnlineset{spread={0.2\normalbaselineskip}}%
5182   \eqnlineset{tagmargin}%
5183   \eqnlineset{tagmarginratio=.334}%
5184   \eqnlineset{tagmarginthreshold=0.5}%
5185   \eqnlineset{leftmargin={\leftmargini}}%
5186   \eqnlineset{padding=0pt}%
5187   \eqnlineset{evadetag}%
5188   \eqnlineset{displayheight=strut}%
5189   \eqnlineset{displaydepth=strut}%
5190   \eqnlineset{shortmode=above}%
5191   \eqnlineset{abovecontmode=noskip}%
5192   \eqnlineset{belowcontmode=long}%
5193   \eqnlineset{aboveparmode=long}%
5194   \eqnlineset{belowparmode=long}%
5195   \eqnlineset{abovetopmode=noskip}%
5196   \eqnlineset{belowtopmode=long}%
5197   \eqnlineset{longskip={0.75\normalbaselineskip
5198     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5199   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5200     plus 0.25\normalbaselineskip}}%
5201   \eqnlineset{belowshortskip={0.0\normalbaselineskip
5202     plus 0.25\normalbaselineskip}}%
5203   \eqnlineset{medskip={0.4\normalbaselineskip
5204     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5205   \eqnlineset{abovecontskip=0pt}%
5206   \eqnlineset{belowcontskip=0pt}%
5207   \eqnlineset{aboveparskip=0pt}%
5208   \eqnlineset{belowparskip=0pt}%
5209   \eqnlineset{abovetopskip=0pt}%
5210   \eqnlineset{belowtopskip=0pt}%
5211   \eqnlineset{abovetagskip={0.25\normalbaselineskip
5212     minus 0.25\normalbaselineskip}}%
5213   \eqnlineset{belowtagskip={0.25\normalbaselineskip
5214     minus 0.25\normalbaselineskip}}%
5215   \eqnlineset{crerror=true}%
5216   \eqnlineset{linesfallback=true}%
5217 }

```



## P.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of `LATEX` and `amsmath` carefully.

### Tools.

`\eql@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```
\eql@provide@moveenv
\eql@provide@movestart
\provide@undefinecmd
\provide@undefineenv
5218 \def\eql@provide@movecmd#1#2{%
5219   \eql@letcs{#1\expandafter}\csname #2\endcsname
5220 }
5221 \def\eql@provide@moveenv#1#2{%
5222   \eql@provide@movecmd{#1}{#2}%
5223   \eql@markline@amsthm@register{#1}%
5224   \ifcsname #2*\endcsname
5225     \eql@provide@movecmd{end#1}{end#2}%
5226   \fi
5227 }
5228 \def\eql@provide@movestart#1#2{%
5229   \eql@provide@moveenv{#1}{#2}%
5230   \ifcsname #2*\endcsname
5231     \eql@provide@moveenv{#1*}{#2*}%
5232   \fi
5233 }
5234 \def\eql@provide@undefinecmd#1{%
5235   \eql@letcs{#1}\undefined
5236 }
5237 \def\eql@provide@undefineenv#1{%
5238   \eql@provide@undefinecmd{#1}%
5239   \eql@provide@undefinecmd{end#1}%
5240 }
```

**Fix Endings for amsmath Environments.** The `amsmath` derived environments forward their ending routines directly to the ending routines for the main environments `gather`, `multline`, `align`, `aligned`. This causes a problem when the main environments are replaced but the derived ones are still used. We fix the potential problem by copying the ending routines of the main environments to the ending routines of the derived environments.

`\eql@amsmath@endfix` Check whether the original forwarding of an ending routine is still in place (other packages or future updates to `amsmath` might change the behaviour). If so, copy the ending routine into place:

```
5241 \def\eql@amsmath@endfix#1#2{%
5242   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
5243   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
5244     \eql@provide@movecmd{end#1}{end#2}%
5245   \fi
5246 }
```

`\eql@amsmath@fixends` Perform the replacement for all `amsmath` environments whenever `amsmath` is loaded:

```
5247 \def\eql@amsmath@fixends{%
5248   \eql@amsmath@after{%
5249     \eql@amsmath@endfix{gather*}{gather}%
5250     \eql@amsmath@endfix{multline*}{multline}%
5251   }
```

```

5251 \eq@amsmath@endfix{align*}{align}%
5252 \eq@amsmath@endfix{flalign}{align}%
5253 \eq@amsmath@endfix{flalign*}{align}%
5254 \eq@amsmath@endfix{alignat}{align}%
5255 \eq@amsmath@endfix{alignat*}{align}%
5256 \eq@amsmath@endfix{xalignat}{align}%
5257 \eq@amsmath@endfix{xalignat*}{align}%
5258 \eq@amsmath@endfix{xxalignat}{align}%
5259 \eq@amsmath@endfix{gathered}{aligned}%
5260 \eq@amsmath@endfix{alignedat}{aligned}%
5261 }
5262 }

```

**Backup amsmath Environments.** We can backup all amsmath environments *env* to *amsenv* so that they can be used in parallel if needed.

**provide@backup@amsmath** Copy an amsmath environment *env* to *amsenv* whenever amsmath is loaded: **TODO:** amsthm

```

5263 \def\eq@provide@backup@amsmath#1{%
5264 \eq@amsmath@after{%
5265 \eq@provide@moveenv{ams#1}{#1}%
5266 \AddToHook{package/amsthm/after}{\eq@provide@movecmd{ams#1@qed}{#1@qed}}%
5267 }%
5268 }

```

**provide@backup@eqref** Copy an eqref to amseqref whenever amsmath is loaded:

```

5269 \def\eq@provide@backup@eqref{%
5270 \eq@amsmath@after{%
5271 \eq@provide@movecmd{amseqref}{eqref}%
5272 }%
5273 }

```

**provide@backup@multlined** The environment *multlined* is supplied by mathtools. We copy it to *amsmultlined* anyway, but whenever mathtools is loaded:

```

5274 \def\eq@provide@backup@multlined{%
5275 \AddToHook{package/mathtools/after}{%
5276 \eq@provide@moveenv{amsmultlined}{multlined}%
5277 }%
5278 }

```

**provide@backup@equation** The L<sup>A</sup>T<sub>E</sub>X environment *equation* is overwritten by several packages to implement their adjustments. Here we cater for adjustments through amsmath, hyperref and the PDF tagging mechanism. Copy *equation* and *equation\** whenever amsmath is loaded. Whenever hyperref is loaded, and amsmath is not yet present, backup the original L<sup>A</sup>T<sub>E</sub>X and hyperref versions of *equation*. If neither hyperref nor amsmath are present, just backup the original L<sup>A</sup>T<sub>E</sub>X *equation*. The PDF tagging mechanism registers *equation* upon `\begin{document}`. We thus need to register all copies of *equation* on our own, so that they can be used with their new names:

```

5279 \def\eq@provide@backup@equation{%
5280 \eq@amsmath@after{%
5281 \eq@provide@moveenv{amsequation}{equation}%
5282 \eq@tagging@register@env{amsequation}%
5283 \eq@provide@moveenv{amsequation*}{equation*}%

```

```

5284 \eql@tagging@register@env{amsequation*}%
5285 \AddToHook{package/amsthm/after}{%
5286 \eql@provide@movecmd{amsequation*@qed}{equation*@qed}}%
5287 }%
5288 \AddToHook{package/hyperref/after}{%
5289 \@ifpackageloaded{amsmath}{}%
5290 \let\latexequation\H@equation
5291 \let\endlatexequation\H@endequation
5292 \eql@tagging@register@env{latexequation}%
5293 \eql@provide@moveenv{hyperrefequation}{equation}%
5294 \eql@tagging@register@env{hyperrefequation}%
5295 \AddToHook{package/amsthm/after}{%
5296 \eql@provide@movecmd{latexequation@qed}{equation@qed}%
5297 \eql@provide@movecmd{hyperequation@qed}{equation@qed}%
5298 }%
5299 }%
5300 }%
5301 \@ifpackageloaded{amsmath}{%}{\@ifpackageloaded{hyperref}{}%
5302 \eql@provide@moveenv{latexequation}{equation}%
5303 \eql@tagging@register@env{latexequation}%
5304 \AddToHook{package/amsthm/after}{%
5305 \eql@provide@movecmd{latexequation@qed}{equation@qed}}%
5306 }%}}%
5307 }

```

e@backup@displaymath **TODO:** describe

```

5308 \def\eql@provide@backup@displaymath{%
5309 \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5310 \AddToHook{package/amsthm/after}{%
5311 \eql@provide@movecmd{latexdisplaymath@qed}{displaymath@qed}}%
5312 }

```

@backup@subequations The amsmath subequations environment is adjusted by hyperref through an environment hook, but this hook gets applied only later at `\begin{document}`. Hence, we need to supply the hook routine to the new routine ourselves:

```

5313 \def\eql@provide@backup@subequations{%
5314 \eql@amsmath@after{%
5315 \eql@provide@moveenv{amssubequations}{subequations}%
5316 }%
5317 \AddToHook{package/hyperref/after}{%
5318 \AddToHook{cmd/amssubequations/before}%
5319 {%
5320 \stepcounter{equation}%
5321 \protected@edef\theHparentequation{\theHequation}%
5322 \addtocounter{equation}{-1}%
5323 }%
5324 \AddToHook{cmd/amssubequations/after}%
5325 {%
5326 \def\theHequation{\theHparentequation\alph{equation}}%
5327 \ignorespaces
5328 }%
5329 }%
5330 }

```

\eql@provide@backup Backup all amsmath environments:

```

5331 \def\eql@provide@backup{%

```

```

5332 \eql@provide@backup@eqref
5333 \eql@provide@backup@equation
5334 \eql@provide@backup@displaymath
5335 \eql@provide@backup@amsmath{gather}%
5336 \eql@provide@backup@amsmath{gather*}%
5337 \eql@provide@backup@amsmath{multline}%
5338 \eql@provide@backup@amsmath{multline*}%
5339 \eql@provide@backup@amsmath{align}%
5340 \eql@provide@backup@amsmath{align*}%
5341 \eql@provide@backup@amsmath{flalign}%
5342 \eql@provide@backup@amsmath{flalign*}%
5343 \eql@provide@backup@amsmath{alignat}%
5344 \eql@provide@backup@amsmath{alignat*}%
5345 \eql@provide@backup@amsmath{xalignat}%
5346 \eql@provide@backup@amsmath{xalignat*}%
5347 \eql@provide@backup@amsmath{xxalignat}%
5348 \eql@provide@backup@amsmath{aligned}%
5349 \eql@provide@backup@amsmath{aligned*}%
5350 \eql@provide@backup@amsmath{alignedat}%
5351 \eql@provide@backup@amsmath{alignedat*}%
5352 \eql@provide@backup@amsmath{gathered}%
5353 \eql@provide@backup@amsmath{gathered*}%
5354 \eql@provide@backup@multlined
5355 \eql@provide@backup@subequations
5356 }

```

**Replacement amsmath Environments.** **TODO:** describe

```

5357 \def\eql@alignat@gobblecol#1{%
5358   \eql@ifnextchar@tight\bgroup{\@firstoftwo{#1}}{#1}}

```

**eql@gathered** (*env.*) Define replacement versions for boxed environments **gathered**, **multlined** and **aligned**  
**eql@multlined** (*env.*) which forward to **equationsbox** with specific presets:

```

eql@aligned (env.)
5359 \newenvironment{eql@gathered}
5360   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5361 \newenvironment{eql@multlined}
5362   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5363 \newenvironment{eql@aligned}
5364   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
5365 \newenvironment{eql@alignedat}
5366   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
5367   {\endequationsbox}

```

**eql@equation** (*env.*) Define replacement versions for display environments **equation**, **gather**, **multline**,  
**eql@gather** (*env.*) **aligned** and derivatives which forward to **equations** with specific presets: **TODO:**  
**eql@multline** (*env.*) **amsmath** at variants would need predefined columns for full operation

```

eql@align (env.)
5368 \newenvironment{eql@equation}
5369   {\eqnaddopt{equation}\equations}{\endequations}
5370 \newenvironment{eql@displaymath}
5371   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
5372 \newenvironment{eql@gather}
5373   {\eqnaddopt{lines}\equations}{\endequations}
5374 \newenvironment{eql@multline}
5375   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5376   {\endequations}
5377 \newenvironment{eql@align}

```

```

5378 {\eqnaddopt{columns}\equations}{\endequations}
5379 \newenvironment{eql@flalign}
5380 {\eqnaddopt{fulllength}\eql@align}{\endequations}
5381 \newenvironment{eql@alignat}
5382 {\eqnaddopt{colsep=off}\eql@xalignat}{\endequations}
5383 \newenvironment{eql@xalignat}
5384 {\eql@alignat@gobblecol\eql@align}{\endequations}
5385 \newenvironment{eql@xxalignat}
5386 {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
5387 \newenvironment{eql@equation*}
5388 {\eqnaddopt{nonumber}\eql@equation}{\endequations}
5389 \newenvironment{eql@gather*}
5390 {\eqnaddopt{nonumber}\eql@gather}{\endequations}
5391 \newenvironment{eql@multline*}
5392 {\eqnaddopt{nonumber}\eql@multline}{\endequations}
5393 \newenvironment{eql@align*}
5394 {\eqnaddopt{nonumber}\eql@align}{\endequations}
5395 \newenvironment{eql@flalign*}
5396 {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
5397 \newenvironment{eql@alignat*}
5398 {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
5399 \newenvironment{eql@xalignat*}
5400 {\eqnaddopt{nonumber}\eql@xalignat}{\endequations}

```

**Install Additional Environments.** The additional environments need to be installed at their intended names which can be adjusted by the user.

**eql@provide@onlyonce** Process arguments for providing a specific environment. #1 describes the environment using the `amsmath` name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the `amsmath` environment in place making sure that the replacement is robust against loading `amsmath` before or after. If #2 equals ‘\*’, just overwrite the `amsmath` environment in place immediately (e.g. within a block in the document body):

```

5401 \def\eql@provide@onlyonce#1#2{%
5402   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
5403   \ifx\eql@tmp\eql@tmpa
5404     \let\eql@tmp\@empty
5405   \fi
5406   \ifx\eql@tmp\@empty
5407     \let\eql@tmp\@undefined
5408     \ifx\@nodocument\relax
5409       \def\eql@tmp{#1}%
5410     \fi
5411     \ifcsname eql@provided@#1\endcsname
5412       \def\eql@tmp{#1}%
5413     \fi
5414     \eql@letcs{eql@provided@#1}\eql@true
5415   \else
5416     \def\eql@tmpa{*}%
5417     \ifx\eql@tmp\eql@tmpa
5418       \def\eql@tmp{#1}%
5419     \fi
5420   \fi
5421 }

```

**\eql@provide@eqref** Provide `\eqref` as the macro #1. We have to check whether #1 is empty or equals `\eqref` or takes the value ‘\*’. If not, we should strip the backslash for further processing. Copy

the macro into place, and copy again when amsmath or mathtools are loaded. Remove definition before amsmath is loaded in the future to avoid a potential error:

```

5422 \def\eq@provide@eqref#1{%
5423   \def\eq@tmp{#1}\def\eq@tmpa{\eqref}%
5424   \ifx\eq@tmp\eq@tmpa
5425     \let\eq@tmp\@empty
5426   \fi
5427   \ifx\eq@tmp\@empty
5428     \eq@provide@onlyonce{eqref}{}%
5429   \else
5430     \def\eq@tmpa{*}%
5431     \ifx\eq@tmp\eq@tmpa
5432       \def\eq@tmp{eqref}%
5433     \else
5434       \edef\eq@tmp{\expandafter\@gobble\string#1}%
5435     \fi
5436   \fi
5437   \ifdefined\eq@tmp
5438     \expandafter\eq@provide@movecmd\expandafter{\eq@tmp}{eq@eqref}%
5439   \else
5440     \eq@amsmath@after{%
5441       \eq@provide@movecmd{eqref}{eq@eqref}%
5442     }%
5443     \AddToHook{package/mathtools/after}{%
5444       \eq@provide@movecmd{eqref}{eq@eqref}%
5445     }%
5446     \eq@provide@movecmd{eqref}{eq@eqref}%
5447     \eq@amsmath@undefine\eqref
5448   \fi
5449 }

```

**\eq@provide@amsmath** Provide one of the amsmath environments and its star variant. Copy into place, and copy again when amsmath or mathtools are loaded. Remove definition before amsmath is loaded in the future to avoid an error:

```

5450 \def\eq@provide@amsmath#1#2{%
5451   \eq@provide@onlyonce{#1}{#2}%
5452   \ifdefined\eq@tmp
5453     \expandafter\eq@provide@movestart\expandafter{\eq@tmp}{eq@#1}%
5454   \else
5455     \eq@amsmath@after{%
5456       \eq@provide@movestart{#1}{eq@#1}%
5457     }%
5458     \AddToHook{package/mathtools/after}{%
5459       \eq@provide@movestart{#1}{eq@#1}%
5460     }%
5461     \eq@provide@movestart{#1}{eq@#1}%
5462     \eq@amsmath@before{\eq@provide@undefineenv{#1}}%
5463     \ifcsname eq@#1*\endcsname
5464       \eq@amsmath@before{\eq@provide@undefineenv{#1*}}%
5465     \fi
5466   \fi
5467 }

```

**\eq@provide@multlined** Provide mathtools environment multlined. Copy into place, and copy again when amsmath or mathtools are loaded. Remove definition before mathtools is loaded in the future to avoid an error:

```

5468 \def\eql@provide@multlined#1{%
5469   \eql@provide@onlyonce{multlined}{#1}%
5470   \ifdefined\eql@tmp
5471     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
5472   \else
5473     \AddToHook{package/mathtools/after}{%
5474       \eql@provide@moveenv{multlined}{eql@multlined}%
5475     }%
5476     \eql@provide@moveenv{multlined}{eql@multlined}%
5477     \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5478       \eql@provide@undefineenv{multlined}}}%
5479   \fi
5480 }

```

**eql@provide@equation** Provide the environment `equation` and its star variant. Copy into place, and copy again when `amsmath` or `hyperref` are loaded. Remove definition of `equation*` before `amsmath` is loaded in the future to avoid an error. When PDF tagging is active, the environment is modified at `\begin{document}` in an undesirable fashion, so copy the definition again:

```

5481 \def\eql@provide@equation#1{%
5482   \eql@provide@onlyonce{equation}{#1}%
5483   \ifdefined\eql@tmp
5484     \expandafter\eql@provide@movestart\expandafter{\eql@tmp}{eql@equation}%
5485   \else
5486     \eql@amsmath@after{%
5487       \eql@provide@movestart{equation}{eql@equation}%
5488     }%
5489     \AddToHook{package/hyperref/after}{%
5490       \ifpackageloaded{amsmath}{}%
5491       \eql@provide@moveenv{equation}{eql@equation}%
5492     }%
5493   }%
5494   \eql@provide@movestart{equation}{eql@equation}%
5495   \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5496   \ifdefined\eql@tagging@on
5497     \AddToHook{begindocument/end}{%
5498       \eql@provide@movestart{equation}{eql@equation}%
5499     }%
5500   \fi
5501   \fi
5502 }

```

**@provide@displaymath** **TODO:** describe

```

5503 \def\eql@provide@displaymath#1{%
5504   \eql@provide@onlyonce{displaymath}{#1}%
5505   \ifdefined\eql@tmp
5506     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@displaymath}%
5507   \else
5508     \eql@provide@moveenv{displaymath}{eql@displaymath}%
5509     \ifdefined\eql@tagging@on
5510       \AddToHook{begindocument/end}{%
5511         \eql@provide@moveenv{displaymath}{eql@displaymath}%
5512       }%
5513     \fi
5514   \fi
5515 }

```

**provide@subequations** Provide the `amsmath` environment `subequations`. Copy into place, and copy again when

amsmath is loaded. hyperref adds a hook to the command which messes up the parsing of optional arguments (even if the hook is emptied). The hook placement happens at `\begin{document}`, so we copy the environment again afterwards. We also remove the hook (after adding an empty hook to avoid errors). Remove definition before amsmath is loaded in the future to avoid an error:

```

5516 \def\eql@provide@subequations#1{%
5517   \eql@provide@onlyonce{subequations}{#1}%
5518   \ifdefined\eql@tmp
5519     \expandafter\eql@provide@moveenv
5520     \expandafter{\eql@tmp}{eql@subequations}%
5521   \else
5522     \eql@amsmath@after{%
5523       \eql@provide@moveenv{subequations}{eql@subequations}%
5524     }%
5525     \AddToHook{package/hyperref/after}{%
5526       \AddToHook{cmd/subequations/before}[hyperref]{}%
5527       \AddToHook{cmd/subequations/after}[hyperref]{}%
5528       \RemoveFromHook{cmd/subequations/before}[hyperref]%
5529       \RemoveFromHook{cmd/subequations/after}[hyperref]%
5530       \AddToHook{begindocument/end}{%
5531         \eql@provide@moveenv{subequations}{eql@subequations}%
5532       }%
5533     }%
5534     \eql@provide@moveenv{subequations}{eql@subequations}%
5535     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
5536   \fi
5537 }

```

`\eql@provide@sqr` Provide the symbolic environment `\[...\]`. Copy into place, and copy again when amsmath is loaded. If PDF tagging is active, some undesired modifications happen at `\begin{document}`, so copy again afterwards:

```

5538 \def\eql@provide@sqr{%
5539   \let\[ \eql@sqr@open
5540   \let\] \eql@sqr@close
5541   \eql@amsmath@after{%
5542     \let\[ \eql@sqr@open
5543     \let\] \eql@sqr@close
5544   }%
5545   \ifdefined\eql@tagging@on
5546     \AddToHook{begindocument/end}{%
5547       \let\[ \eql@sqr@open
5548       \let\] \eql@sqr@close
5549     }%
5550   \fi
5551 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```

5552 \def\eql@provide@ang{%
5553   \let\< \eql@ang@open
5554   \let\> \eql@ang@close
5555 }

```

**Interface.**



`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

5556 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5557 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5558 \eql@define@key{provide}{gather}[]{\eql@provide@amsmath{gather}{#1}}
5559 \eql@define@key{provide}{multline}[]{\eql@provide@amsmath{multline}{#1}}
5560 \eql@define@key{provide}{align}[]{\eql@provide@amsmath{align}{#1}}
5561 \eql@define@key{provide}{flalign}[]{\eql@provide@amsmath{flalign}{#1}}
5562 \eql@define@key{provide}{alignat}[]{\eql@provide@amsmath{alignat}{#1}}
5563 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsmath{xalignat}{#1}}
5564 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsmath{xxalignat}{#1}}
5565 \eql@define@key{provide}{aligned}[]{\eql@provide@amsmath{aligned}{#1}}
5566 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsmath{alignedat}{#1}}
5567 \eql@define@key{provide}{gathered}[]{\eql@provide@amsmath{gathered}{#1}}
5568 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5569 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5570 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
5571 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
5572 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
5573 \eql@define@key{provide}{tagform}[]{%
5574   \def\tagform@##1{\maketag@@@{\eql@tags@tagform{#1}}}}
5575 \eql@define@key{provide}{maketag}[]{%
5576   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}
```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```

5577 \newcommand{\eqnlinesprovide}[1]{%
5578 (dev)\eql@dev@start\eqnlinesprovide
5579   \eql@setkeys{provide}{#1}%
5580   \ignorespaces
5581 }
```

## P.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
5582 \let\eql@error@packageoption@gobble
```

Declare math layout options `leqno` and `fleqn` for common L<sup>A</sup>T<sub>E</sub>X classes:

```

5583 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
5584 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
5585 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

5586 \eql@defaults@eqnlines
5587 \eql@mode@columns
5588 \eql@mode@aligned
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the T<sub>E</sub>X conditional parsing mechanism:

```

5589 \ifdefined\tagsleft@true\else
5590   \expandafter\newif\csname iftagsleft@\endcsname
```

```

5591 \fi
5592 \ifdefined\@fleqntrue\else
5593   \expandafter\newif\csname if@fleqn\endcsname
5594 \fi

```

Import amsmath switches leqno as tagsleft and fleqn as left:

```

5595 \eq@amsmath@after{%
5596   \ifnum\eq@provide@opt@env=\tw@
5597     \iftagsleft@
5598       \eqnlineset{tags=left}%
5599     \else
5600       \eqnlineset{tags=right}%
5601     \fi
5602     \if@fleqn
5603       \eqnlineset{layout=left}%
5604     \else
5605       \eqnlineset{layout=center}%
5606     \fi
5607   \fi
5608 }

```

Process package options:

```

5609 \ProcessOptions

```

**@error@packageoption** Enable error message for exclusive package options:

```

5610 \def\eq@error@packageoption#1{%
5611   \eq@error{may only use '#1' as a package option}%
5612 }

```

Make the ending statements for amsmath environments independent if desired, so that they may be overwritten individually:

```

5613 \ifdefined\eq@provide@opt@amsmathends\eq@amsmath@fixends\fi

```

Backup all amsmath environments that may be overwritten to `ams...`. This will happen before any replacements:

```

5614 \ifdefined\eq@provide@opt@backup\eq@provide@backup\fi

```

Provide native L<sup>A</sup>T<sub>E</sub>X environment `equation` and symbolic shortcut `\[...\]` if desired:

```

5615 \ifnum\eq@provide@opt@env>\z@
5616   \eqnlinesprovide{equation,sqr,displaymath}
5617 \fi

```

Provide amsmath equation environments if desired:

```

5618 \ifnum\eq@provide@opt@env=\tw@
5619   \eqnlinesprovide{%
5620     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
5621     multlined,gathered,aligned,alignedat,%
5622     subequations}
5623 \fi

```

Provide symbolic shortcut `\<...\>` if desired:

```

5624 \ifdefined\eq@provide@opt@ang\eqnlinesprovide{ang}\fi

```

Provide equation reference `\eqref` if desired:

```

5625 \ifdefined\eq@provide@opt@eqref\eqnlinesprovide{eqref}\fi

```