

# 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`

2025/04/30, v0.8

<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 . . . . .	7
2.3	Horizontal Adjustment . . . . .	10
2.4	Punctuation . . . . .	14
2.5	Math Classes at Alignment . . . . .	15
2.6	Vertical Spacing . . . . .	16
2.7	Further Environments and Features . . . . .	18
2.8	General Options . . . . .	20
2.9	Feature Selection and Package Options . . . . .	21
<b>3</b>	<b>Information</b>	<b>21</b>
3.1	Copyright . . . . .	21
3.2	Credits . . . . .	21
3.3	Files and Installation . . . . .	22
3.4	Related CTAN Packages . . . . .	22
3.5	Feature Suggestions . . . . .	23
3.6	Revision History . . . . .	23
<b>A</b>	<b>Implementation</b>	<b>24</b>

<b>B</b>	<b>General Support</b>	<b>25</b>
B.1	Development Messages . . . . .	25
B.2	Supporting Definitions . . . . .	25
B.3	Dollardollar Abstraction . . . . .	25
B.4	Look-Ahead in Alignment . . . . .	25
B.5	Error Messages . . . . .	27
B.6	amsmath Integration . . . . .	28
B.7	PDF Tagging Support . . . . .	28
<b>C</b>	<b>Parameters and Registers</b>	<b>30</b>
C.1	Parameters . . . . .	30
C.2	Registers . . . . .	33
C.3	Hooks . . . . .	35
<b>D</b>	<b>Features</b>	<b>35</b>
D.1	Punctuation . . . . .	35
D.2	Math Classes at Alignment . . . . .	37
D.3	Framed Cells . . . . .	38
D.4	Alternative Content Description . . . . .	38
<b>E</b>	<b>Equation Numbering</b>	<b>38</b>
E.1	Supporting Definitions . . . . .	38
E.2	Schemes . . . . .	39
E.3	Interface . . . . .	41
E.4	Integration . . . . .	45
E.5	Component Display . . . . .	47
E.6	Tag Composition . . . . .	51
<b>F</b>	<b>Subequation Numbering</b>	<b>52</b>
F.1	Definitions . . . . .	53
F.2	Environment . . . . .	54
F.3	Subequation Scheme . . . . .	55
<b>G</b>	<b>Display Equations Support</b>	<b>56</b>
G.1	Display Breaks . . . . .	56
G.2	Vertical Spacing Support . . . . .	57
G.3	Entry and Exit . . . . .	58
G.4	Stack . . . . .	63
<b>H</b>	<b>Multi-Line Support</b>	<b>65</b>
H.1	Measure Support . . . . .	65
H.2	Line Breaks . . . . .	66
H.3	Intertext . . . . .	66
<b>I</b>	<b>Column Placement</b>	<b>69</b>
I.1	Supporting Definitions . . . . .	69
I.2	Shape Schemes . . . . .	69
I.3	Width Data . . . . .	72
I.4	Best Line Selection . . . . .	76
I.5	Tag Margin . . . . .	77
I.6	Single Column . . . . .	77
I.7	Multiple Columns . . . . .	79
<b>J</b>	<b>Single Column Arrangement</b>	<b>84</b>
J.1	Supporting Definitions . . . . .	84

J.2	Arrangement Methods . . . . .	85
J.3	Central Alignment . . . . .	86
J.4	Left Alignment . . . . .	88
J.5	Right Alignment . . . . .	89
<b>K</b>	<b>Equations Box Environment</b>	<b>90</b>
K.1	Line Breaks . . . . .	90
K.2	Stacked Mode . . . . .	91
K.3	Aligned Mode . . . . .	92
K.4	Main . . . . .	93
K.5	Environment . . . . .	95
<b>L</b>	<b>Single-Line Equation</b>	<b>96</b>
L.1	Native Mode . . . . .	96
L.2	Print . . . . .	96
<b>M</b>	<b>Multi-Line with Single Column</b>	<b>97</b>
M.1	Measure . . . . .	97
M.2	Column Placement . . . . .	99
M.3	Print . . . . .	99
<b>N</b>	<b>Multi-Line with Multiple Columns</b>	<b>100</b>
N.1	Support . . . . .	101
N.2	Measure . . . . .	101
N.3	Columns Placement . . . . .	104
N.4	Print . . . . .	104
<b>O</b>	<b>Interface</b>	<b>107</b>
O.1	Scanning the Equation Body . . . . .	107
O.2	Options Processing . . . . .	111
O.3	Single-Line Main . . . . .	112
O.4	Multi-Line Main . . . . .	113
O.5	Equations Environment . . . . .	115
<b>P</b>	<b>Options</b>	<b>117</b>
P.1	Selection Tools . . . . .	117
P.2	Declaration Code . . . . .	119
P.3	Options Declarations . . . . .	119
P.4	Parameter Presets . . . . .	128
P.5	Component Selection . . . . .	130
P.6	Global and Package Options . . . . .	138

## 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  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  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  $\text{\LaTeX}$  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  $\text{\TeX}$  era and early  $\text{\LaTeX}$  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  $\text{\LaTeX}$  2<sub>ε</sub> code without using the `expl3` layer.

The package represents a stand-alone implementation of an equations environment which is largely compatible with the established  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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 v0.8:** 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

$$\backslash\text{usepackage}\{\text{eqnlines}\}$$

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.*) 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 `\[...\]`:

$$\text{single line}$$

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

$$\begin{array}{c} \text{stacked line 1} \\ \text{stacked line 2} \\ \text{stacked line 3} \\ \text{aligned line 4} \end{array}$$

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

$$\begin{array}{cc} \begin{array}{cc} 1a\text{-L} & 1a\text{-R} \\ 2a\text{-L} & 2a\text{-R} \\ & 3a\text{-R} \end{array} & \begin{array}{cc} 1b\text{-L} & 1b\text{-R} \\ & 2b\text{-R} \end{array} \end{array}$$

The environment `equations` accepts a comma-separated list of optional parameters ‘`[opts]`’:

$$\begin{array}{l} \backslash\text{begin}\{\text{equations}\}[opts]_ \\ \dots \\ \backslash\text{end}\{\text{equations}\} \end{array}$$

Furthermore, the environment accepts some modifiers (like the star modifier ‘`*`’ for many other  $\text{\LaTeX}$  macros) which will be explained further below. These follow the scheme `{ !t~ !t* !t! !o !e{@} }` according to the syntax of `\NewDocumentCommand`.

<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.

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.

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

<code>lines</code> ( <i>key</i> )	<code>columns</code> ( <i>key</i> )	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>~</code>	<code>align</code> , <code>al</code> , <code>col</code> , <code>@</code>
		symbolic	<code>\[...\]</code>	<code>\&lt;~...\&gt;</code>	<code>\&lt;...\&gt;</code>
		amsmath 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 &amp; \phi&amp;=\arccos x \\\&amp;=(z+z^{-1})/2 &amp; \&amp;=-i\log z</code>	$x = \cos \phi$	$\phi = \arccos x$	(4)
<code>\end{equations}</code>	$= (z + z^{-1})/2$	$= -i \log z$	(5)

`\[...\]` 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.

`~` (*key*) Here, the tilde ‘`~`’ in `\<~...\>` is a modifier character which acts as a short form for the

`sqropt` (*key*) optional argument `lines` selecting the lines mode. Both short forms can be customised by  
`angopt` (*key*) setting default arguments via the global 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 \\\&amp;=(z+z^{-1})/2 &amp; \&amp;=-i\log z</code>	$x = \cos \phi$	$\phi = \arccos x$	
<code>\&gt;</code>	$= (z + z^{-1})/2$	$= -i \log z$	

---

```
\eqnlineset{sqropt={donumber}}
\l x=\cos\phi \l
```

$$x = \cos \phi \quad (6)$$


---

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

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

\begin{equation}
x=\cos\phi
\end{equation}

$$x = \cos \phi \quad (7)$$


\begin{gather}
x=\cos\phi \ll \phi=\arccos x
\end{gather}

$$x = \cos \phi \quad (8)$$


$$\phi = \arccos x \quad (9)$$


\begin{align}
x&=\cos\phi \ \& \ \phi&=\arccos x \ll \quad x = \cos \phi \quad \phi = \arccos x \quad (10) \\
&=(z+z^{-1})/2 \ \& \ \&=-i\log z \quad = (z + z^{-1})/2 \quad = -i \log z \quad (11)
\end{align}

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

$$x = \cos \phi$$


$$\phi = \arccos x$$

```

---

## 2.2 Numbering

`numberline` (*key*) 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>none</code>	<code>n</code>		all lines, preset off
<code>all</code>	<code>a</code>	individual lines	all lines
<code>sub</code>	<code>s</code>		subequations (a, b, c, ...)
<code>first</code>	<code>f</code>		first line
<code>last</code>	<code>l</code>		last line
<code>middle</code>	<code>m</code>	single number	middle 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>here</code>	<code>h</code>		line indicated by <code>\numberhere</code>
<code>best</code>	<code>*</code>		line with most available space

---

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

<b>none:</b>	<b>all:</b>	<b>sub:</b>
$x = \cos \phi$	$x = \cos \phi$ (12)	$x = \cos \phi$ (16a)
$= (z + z^{-1})/2$	$= (z + z^{-1})/2$ (13)	$= (z + z^{-1})/2$ (16b)
$\phi = \arccos x$	$\phi = \arccos x$ (14)	$\phi = \arccos x$ (16c)
$= -i \log z$	$= -i \log z$ (15)	$= -i \log z$ (16d)
<b>first:</b>	<b>middle:</b>	<b>last:</b>
$x = \cos \phi$ (17)	$x = \cos \phi$	$x = \cos \phi$ (20)
$= (z + z^{-1})/2$	$= (z + z^{-1})/2$ (18)	$= (z + z^{-1})/2$
$\phi = \arccos x$	$\phi = \arccos x$	$\phi = \arccos x$
$= -i \log z$	$= -i \log z$	$= -i \log z$ (19)

**bestlineauto** (*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 **bestlineauto=bool**.

**\nonumber** Numbering can be turned on and off (for individual lines or 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**

**donumber** (*key*) (**nn='\*' or dn='!' for short) or by **number=bool** with *bool* either **on** or **off** (among several**

**number** (*key*) alternative forms). Alternatively the number can be switched by using modifiers (which

**nn,\*** (*key*) cannot be used in conjunction with optional arguments [...]):

**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.

---

```

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

```

$$\begin{aligned}
 x &= \cos \phi \\
 &= (z + z^{-1})/2 \\
 \phi &= \arccos x \\
 &= -i \log z
 \end{aligned}
 \tag{21}$$

```

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

```

$$\begin{aligned}
 x &= \cos \phi \\
 &= (z + z^{-1})/2 \\
 \phi &= \arccos x \\
 &= -i \log z
 \end{aligned}
 \tag{23}$$

```

\eqnlineset{numberline=last}
\<! x &= \cos\phi \\
\phi &= \arccos x \>

```

$$\begin{aligned}
 x &= \cos \phi \\
 \phi &= \arccos x
 \end{aligned}
 \tag{25}$$



<code>\eqnlineset{angopt=donumber}</code>	$x = \cos \phi$	
<code>\&lt;* x &amp;= \cos\phi \\\</code>	$\phi = \arccos x$	
<code>\phi &amp;= \arccos x \&gt;</code>		
<code>\begin{equations}</code>		
<code>  x &amp;= \cos\phi \numbernext \\\</code>	$x = \cos \phi$	
<code>    &amp;= (z+z^{-1})/2 \\\</code>	$= (z + z^{-1})/2$	(26)
<code>\phi &amp;= \arccos x \numbernext \\\</code>	$\phi = \arccos x$	
<code>    &amp;= -i\log z</code>	$= -i \log z$	(27)
<code>\end{equations}</code>		
<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>	$\phi = \arccos x$	(28)
<code>\phi &amp;= \arccos x \numberhere \\\</code>	$= -i \log z$	
<code>    &amp;= -i\log z</code>		
<code>\&gt;</code>		
<code>\eqnlineset{numberline=first}</code>		
<code>\&lt;!</code>	$x = \cos \phi$	
<code>  x &amp;= \cos\phi \numbernext \\\</code>	$= (z + z^{-1})/2$	(29)
<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>		
<code>\&gt;</code>		

---

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

`\label[name]{label}`      and      `\tag*[{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. 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]* or via the modifier `@{label}` (which may *follow* further op-  
`labelname (key)` tional arguments):  
`@ (key)`

`label={label},      tag[*]={tag},      labelname={name},      \[@{label}... \]`

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.

`\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`  
`\tagboxed`

decorated number in a box and `\tagboxed` combines the two.

`tagbox` (*key*) The typesetting of equation numbers and tags passes through two macros, one which defines the layout and another one which adds a decoration by parentheses. These two methods 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.

---

```

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

```

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

[30]

---

## 2.3 Horizontal Adjustment

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

`left` (*key*)

---

```

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

```

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

```

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

```

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

---

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

`tagsleft` (*key*)

---

```

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

```

$x = \cos \phi$  (31)  
 $= (z + z^{-1})/2$  (32)  
 $\phi = \arccos x$  (33)  
 $= -i \log z$  (34)

```

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

```

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

---

**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 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}
\[\! \framebox[4em]{} \] (39)
\[\! \framebox[8em]{} \] (40)
\[\! \framebox[12em]{} \] (41)
\[\! \framebox[16em]{} \] (42)
```

---

**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
\>
```

---

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

---

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

---

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

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

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

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

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
<b>default</b>	<b>def</b>	uniform	default
<b>left</b>	<b>l</b>		left
<b>center</b>	<b>c</b>	uniform	central
<b>right</b>	<b>r</b>		right
<b>first</b>	<b>indent, rc</b>	first/rest	first line indented
<b>hanging</b>	<b>outdent, lc</b>	first/rest	first line hanging
<b>steps</b>	<b>lcr</b>	first/intermediate/last	left/centre...centre/right

Note that the **steps** shape comes to use in the **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>
$x = \cos \phi$	$x = \cos \phi$	$x = \cos \phi$
$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$	$x = (z + z^{-1})/2$
$\phi = \arccos x$	$\phi = \arccos x$	$\phi = \arccos x$
$\phi = -i \log z$	$\phi = -i \log z$	$\phi = -i \log z$

**\shoveleft** The alignment preset can be adjusted for individual lines by the macros:  
**\shovecenter**  
**\shoveright** **\shoveleft|center|right[\*][dimen]**,

In contradistinction to **amsmath**, these macros do not require to specify the cell contents as their argument (but there is no harm in doing so). The macro accept an optional argument **indent (key)** *[dimen]* specifying a variable amount of shift. They also accept the modifiers ‘\*’ or ‘!’ for indentation or hanging indentation by the standard indentation amount (**indent=2em**).  
**\shoveby** Furthermore, **\shoveby[\*]{dimen}** shifts the line by the amount *dimen*, where the star

variant resets the shift amount first.

**padding** (*key*) Note that (hanging) indentation requires to add some padding around the equations block  
**padleft** (*key*) via the optional argument `padding|padleft|padright[={dimen}]`. The value `indent` sets  
**padright** (*key*) 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.

---

```

\eqnlineset{indent=2em,pad=5em}
\<~
\shoveleft   \framebox[5em]{left}  \
\shoveleft*  \framebox[5em]{indent} \
\shovecenter \framebox[5em]{center} \
\shoveright  \framebox[5em]{right}
\>
\eqnlineset{layout=left}
\eqnlineset{leftmargin=2em}
\eqnlineset{indent=2em}
\<~
\shoveleft!  \framebox[5em]{outdent} \
\shoveleft   \framebox[5em]{left}   \
\shoveleft*  \framebox[5em]{indent}  \
\shoveright  \framebox[5em]{right}
\>

```

---

**alignshrink** (*key*) Finally, we note that within single and stacked equations, very long equations that do not  
**tagshrink** (*key*) fit the available horizontal space are subject to shrinking attempts. In other words, T<sub>E</sub>X will  
**alignbadness** (*key*) attempt to shrink the glue contained in the equation line to make it fit. This shrinking can be  
**tagbadness** (*key*) controlled by the two parameters `alignbadness` and `tagbadness` accepting integer values.  
They are used towards determining whether to shift away from the intended alignment or  
whether to raise or lower the equation tag, respectively. Small values prevent shrinking and  
higher values allow for more compression.

---

```

\<~!
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 \
\>

```

$$x + x \quad (47)$$

$$x + x + x + x \quad (48)$$

$$x + x + x + x + x \quad (49)$$

$$x + x + x + x + x + x + x \quad (50)$$

$$x + x + x + x + x + x + x + x \quad (51)$$

$$x + x + x + x + x + x + x + x + x + x \quad (52)$$


---

**mintagsep** (*key*) If the available space on a line does not suffice to place both the equation and its tag (with  
**\raisetag** a minimum separation of `mintagsep`; default is `0.5em`), a tag will automatically be lowered  
or raised (depending on whether it is placed on the right or left). The macro `\raisetag`  
may be used to fine-tune the vertical placement (applies only if the tag is already shifted  
due to lack of space).

---

```

\![ \phi = -\int \frac{\mathrm{d}x}{\sqrt{1+x^2}} \ ]
\![ x = \frac{\partial}{\partial \phi} \sin \phi \ ]
\raisetag{0.45\baselineskip} \ ]

```

$$\phi = - \int \frac{dx}{\sqrt{1+x^2}} \quad (53)$$

$$x = \frac{\partial}{\partial \phi} \sin \phi \quad (54)$$


---

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

---

<code>\[ \indicate{line width} \]</code>	<div>line width</div>
<code>\[[margin=2em] \indicate{reduced} \]</code>	<div>reduced</div>
<code>\begin{itemize}</code>	
<code>\item first level</code>	• first level
<code>\[ \indicate{default width} \]</code>	<div>default width</div>
<code>\[[marginleft=0pt]</code>	
<code>\indicate{full width} \]</code>	<div>full width</div>
<code>\end{itemize}</code>	

---

## 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 author are willing to afford. A contributing factor is that punctuation marks are harder to spot alongside equation context and somewhat out of place anyway.

`\eqnpunctdefault` 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:

`\eqnpunct`  
`punct` (*key*)

```
\eqnpunctdefault{punct} \eqnlineset{punct={punct}}
\eqnpunct{punct}        \eqnadopt{punct={punct}}
\[[punct={punct}] ... \]
```

The former two forms set and enable a default punctuation mark; the middle two forms set the punctuation mark for the next equation environment in line; the final form applies to the equation environment only. For example, one might declare ‘`\eqnpunctdefault.`’ 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{}` and `[punct{}]` eliminate a preset punctuation.

---

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

---

`\eqnpunctapply` In situations, where the punctuation must appear before the end of the block, e.g. before a “q.e.d.”, it can be invoked manually by `\eqnpunctapply`.

`punctsep` (*key*) For convenience, one may also specify a desired space (or any other code sequence) preceding

---

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

the punctuation by `[punctsep={sep}]`, e.g. `sep=\`, or `sep=\_`.

`\eqnpunctcol` For multi-line equations, there are two further levels of default punctuation for terminating  
`\eqnpunctline` columns and lines which are specified via the macros `\eqnpunctcol` and `\eqnpunctline` or  
`punctcol (key)` the optional arguments `punctcol` and `punctline`. A punctuation item may also be handed  
`punctline (key)` on to the next lower level of punctuation via the starred forms `punct*` and `punctline*`.

---

```

\eqnpunctcol, \eqnpunctline;
\eqnpunctdefault.
\< x &= \cos\phi &          x = \cos \phi,          \phi = \arccos x;
\phi &= \arccos x \\\      x = (z + z^{-1})/2,    \phi = -i \log z.
  x &= (z+z^{-1})/2 &
\phi &= -i\log z \>

```

---

## 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 T<sub>E</sub>X'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:

---

```

\[class=ampeq]
a &= b+c \\\      a = b + c
  &= d-e \\\      = d - e
  &\mathrel{=}\phantom{=} +f      + f
\>
\[class=eqamp]
a =& b+c \\\      a = b + c
  =& d-e \\\      = d - e
  & +f              + f
\>

```

---

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

`classlead (key)`

$$\text{classout}=\{class\}, \quad \text{classin}=\{class\}, \quad \text{classlead}=\{class\}.$$

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:

$$\begin{array}{c|c} \square\square\square a & \underline{a-out} \quad \underline{in-b} \quad \square\square\square b \\ & \underline{lead-c} \quad \square\square\square c \end{array}$$

## 2.6 Vertical Spacing

Display equations in  $\text{\TeX}$  are considered to be part of the surrounding text. Hence, the vertical spacing depends on the surrounding text, in particular on the width and depth of the last line of text. 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 of equations to the surrounding text is a combination of several aspects:

First,  $\text{\TeX}$  inserts some interline spacing according to its rules. The amount depends on the depth/height of the surrounding text and the height/depth of the math content. The former typically takes rather uniform values, whereas the latter can range wildly with the context (plain equations vs. fractions and matrices). As 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` (*key*) and `displaydepth` (*key*) which default to the dimensions of a strut.

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  $\text{\TeX}$  feature for all single- and multi-line equation environments.

---

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

---

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

<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:	and short:
<code>\[[long] \mbox{forced long} \]</code>	forced long
following line	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 <code>\par</code>	some text
<code>\[ \mbox{par} \]</code>	
<code>\[ \mbox{cont} \]</code>	par
<code>\end{minipage}\hrule</code>	cont

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*) The key `par` controls whether the equation environments end in horizontal mode (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`.

`...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`.

The package also maintains several global vertical space settings `aboveposskip` and `...skip` (*key*) `belowposskip` (sometimes `posskip` for both):

<code>...posskip</code>	both	description
<code>...long...</code>	<code>longskip</code>	regular amount of glue
<code>...short...</code>	–	reduced glue for short equations
<code>...cont...</code>	–	glue when issued from an empty <code>\noindent</code> 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>	<code>medskip</code>	medium amount of glue
<code>...tag...</code>	<code>tagskip</code>	glue for outer raised/lowered tags
<code>...medtag...</code>	<code>medtagskip</code>	glue for outer raised/lowered tags with medium glue
<code>...partag...</code>	–	glue for outer raised/lowered tags with par skip

`...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

`spread (key)` Likewise, the spacing between the lines of a multi-line equation environment can be adjusted

`strut (key)` via `spread={dimen}` which defaults to `\jot=3pt`. In addition, 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`.

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

---

```

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

$\Longrightarrow\!<[shape=l,frame]
  x = \cos\phi \ &
\phi = \arccos x \ \
  x = (z+z^{-1})/2 \ &
\phi = -i\log z
\Longleftarrow$

```

---


$$\left\{ \begin{array}{l} x = \cos \phi \\ \phi = \arccos x \end{array} \right\}$$

$$\Rightarrow \boxed{\begin{array}{ll} x = \cos \phi & \phi = \arccos x \\ x = (z + z^{-1})/2 & \phi = -i \log z \end{array}} \Leftarrow$$


---

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

$$\text{subeqtemplate}=\{\#1\backslash\text{alph}\{\#2\}\}$$


---

```

\eqnlineset
  {subeqtemplate=\#1-\roman{\#2}}
\begin{subequations}
\left[ x = \cos\phi \right] \qquad \text{and} \qquad x = \cos \phi \qquad (55\text{-i})
\text{and} \qquad \phi = \arccos x \qquad (55\text{-ii})
\left[ \phi = \arccos x \right]
\end{subequations}

```

---

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

---

```

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

```

---

**\framecell** **Framed Cells.** The package allows to frame cells of an equation block via issuing a simple command within the cell:

$$\backslash\text{framecell}[cmd]$$

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.

---

<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]{\mbox{important} \eqnpunct! \}</code>	important!
<code>\[\framecell[\fcolorbox{white}{yellow}]{\mbox{highlight}}\]</code>	highlight

---

**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/ <code>amsmath</code> (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>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	type	description
<code>equation</code>	bool	provide/overwrite <code>equation</code> and <code>\[...\]</code>
<code>amsmath</code>	bool	provide/overwrite <code>amsmath</code> environments and macros
<code>amsmathends</code>	bool	patch <code>amsmath</code> environments with individual endings
<code>backup</code>	bool	backup original <code>amsmath</code> environments as <code>ams...</code>
<code>ang</code>	bool	provide <code>\&lt;...\&gt;</code>
<code>eqref</code>	bool	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 latex package `amsmath`: Copyright © 1995, 2000, 2013 American Mathematical Society; 2016–2024 LaTeX Project and American Mathematical Society.

This work may be distributed and/or modified under the conditions of the 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  $\text{\LaTeX}$  was done in 1988–1989 by Frank Mittelbach and Rainer Schöpf. In 1994 David M. Jones added the support for flush-left layout and did extensive improvements to the align family of environments and to the equation number handling in general. Michael Downes at the AMS served as coordinator for the efforts of Mittelbach, Schöpf, and Jones, and has contributed various bug fixes and additional refinements over time. Since 2016, the package has been maintained by the  $\text{\LaTeX}$  Project with contributions by the above and David Carlisle.

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

- The implementations of `split` and methods unrelated to multi-line equations and equation numbering have been dropped.
- Code has been restructured, macros have been renamed and extended.
- Numbering and horizontal adjustment schemes have been unified and extended.
- Options for math classes surrounding the alignment have been added.
- A punctuation scheme has been added.
- Vertical spacing has been redesigned.
- Optional parameters have been added to environments.
- Various configuration options and layout settings have been added.
- Cooperation with `hyperref`, `showkeys` and `amsmath` has been included into the package.

### 3.3 Files and Installation

The package consists of the files:

<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) $\text{\LaTeX}$  on `eqnlines.dtx` to compile the manual `eqnlines.pdf` (this file).
- Run  $\text{\LaTeX}$  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  $\text{\LaTeX}$  distribution, e.g. `texmf-root/tex/latex/eqnlines`.

### 3.4 Related CTAN Packages

The package is related to other packages available at CTAN:

- This package uses the package `keyval` to process the options for the package, environments and macros. Compatibility with the `keyval` package has been tested with v1.15 (2022/05/29).

- This package reproduces the math environments functionality of the package `amsmath`. The present code is based on `amsmath` v2.17t (2024/11/05). Compatibility with the `amsmath` package is maintained whether `eqnlines` is loaded before or after `amsmath`. By default, `eqnlines` overwrites most math environments of `amsmath` with its own implementations. It can also preserve them as `ams...` if needed. Alternatively, `eqnlines` may assign individual names to the maths environments and preserve the ones of `amsmath`. The other features provided by `amsmath` can be used.
- The package `mathtools` is a popular extension of the `amsmath` package. This package incorporates some of the features and improvements provided by the `mathtools` package. Compatibility with the `mathtools` package has been tested with v1.31 (2024/10/04), and it is maintained whether `eqnlines` is loaded before or after `mathtools`. Some features like emphasising equations via `empheq` does 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 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).

### 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

**v0.8:** 2025/04/30

- added framed cells (`\framecell`)
- added automatic best line selection for tag placement (`best` and `bestlineauto`)
- 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

## 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 \@currrent}}
6 <dev>\def\eql@dev@leaveenv{\eql@dev{ leaving \@currrent}}
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
```

### 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:

```
10 \ifdefined\dollar@begin
11   \def\eql@dollar@begin{\dollar@begin}
12   \def\eql@dollar@end{\dollar@end}
13 \else
14   \def\eql@dollar@begin{$$$}
15   \def\eql@dollar@end{$$$}
16 \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.

`\eql@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 `\eql@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:

```

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

```

`\eq@atxii` 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:

```

30 \begingroup
31   \makeatother
32   \let\tmp=@%
33   \makeatletter
34   \global\let\eq@atxii\tmp
35 \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 [...]:

```

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

```

`\eq@spbgrou` The second challenge is addressed by enclosing the look-ahead in spurious groups<sup>3</sup> which  
`\eq@spegroup` protect against triggering ‘&’. The macros `\eq@spbgrou` and `\eq@spegroup` open and  
`\eq@srbgroup`  
`\eq@sregroup`

<sup>3</sup>See <https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3040>,

close a spurious group. For some reason, the look-ahead mechanism requires further protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end. These adjustments are included in the macros `\eq@srbbgroup` and `\ers@speggroup`:

```
56 \def\eq@spbbgroup{\iffalse{\fi\ifnum0='}\fi}
57 \def\eq@speggroup{\ifnum0='{ \fi\iffalse}\fi}
58 \def\eq@srbbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
59 \def\eq@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}
```

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

```
60 \long\def\eq@ampprotect#1#2{\eq@srbbgroup#1{\eq@sregroup#2}}
61 \long\def\eq@ampprotecttwo#1#2#3{%
62   \eq@srbbgroup#1{\eq@sregroup#2}{\eq@sregroup#3}}
```

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

```
63 \def\eq@ifnextchar@loose@ampsafe#1{%
64   \eq@ampprotecttwo{\eq@ifnextchar@loose#1}}
65 \def\eq@ifnextchar@tight@ampsafe#1{%
66   \eq@ampprotecttwo{\eq@ifnextchar@tight#1}}
67 \def\eq@ifstar@loose@ampsafe{\eq@ampprotecttwo\eq@ifstar@loose}
68 \def\eq@ifstar@tight@ampsafe{\eq@ampprotecttwo\eq@ifstar@tight}
69 \def\eq@testopt@loose@ampsafe{\eq@ampprotect\eq@testopt@loose}
70 \def\eq@testopt@tight@ampsafe{\eq@ampprotect\eq@testopt@tight}
71 \def\eq@teststaropt@loose@ampsafe{\eq@ampprotecttwo\eq@teststaropt@loose}
72 \long\def\eq@teststaropt@tight@ampsafe{%
73   \eq@ampprotecttwo\eq@teststaropt@tight}
```

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

```
74 \let\eq@ifnextchar@org\@ifnextchar
75 \let\eq@ifstar@org\@ifstar
76 \let\eq@testopt@org\@testopt
77 \def\eq@amprevert{%
78   \let\@ifnextchar\eq@ifnextchar@org
79   \let\@testopt\eq@testopt@org
80   \let\@ifstar\eq@ifstar@org
81 }
82 \def\eq@amproof{%
83   \let\@ifnextchar\eq@ifnextchar@loose@ampsafe
84   \let\@testopt\eq@testopt@loose@ampsafe
85   \let\@ifstar\eq@ifstar@loose@ampsafe
86 }
```

## B.5 Error Messages

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

```
87 \def\eq@error#1{\PackageError{eqnlines}{#1}{}}
88 \def\eq@warning{\PackageWarning{eqnlines}}
```

<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>.

`\eqLError@nomathmode` Error messages concerning math mode:

```
\eqLError@mathmode
89 \def\eqLError@nomathmode#1{\eqLError{#1 allowed only in math mode}}
90 \def\eqLError@mathmode#1{\eqLError{#1 allowed only in paragraph mode}}
```

`\eqLWarn@label@unused` Warning messages concerning unused and multiply declared labels and tags:

```
\eqLWarn@label@multiple
\eqLWarn@tag@unused
\eqLWarn@tag@multiple
\eqLWarn@labelname@unused
\eqLWarn@labelname@multiple
91 \def\eqLWarn@label@unused{\eqLWarning{Unused equation \string\label:
92   label '\eqLNextlabel' will be lost}}
93 \def\eqLWarn@label@multiple#1#2{\eqLWarning{%
94   Multiple equation \string\label's:
95   previous label '#1' will be lost in favor of '#2'}}
96 \def\eqLWarn@labelname@unused{\eqLWarning{Unused equation label name:
97   name declaration will be lost}}
98 \def\eqLWarn@labelname@multiple{\eqLWarning{Multiple equation label names:
99   previous name declaration will be lost}}
100 \def\eqLWarn@tag@unused{\eqLWarning{Unused equation \string>tag:
101   tag declaration will be lost}}
102 \def\eqLWarn@tag@multiple{\eqLWarning{Multiple equation \string>tag's:
103   previous tag declaration will be lost}}
```

## B.6 amsmath Integration

`\eqLamsmath@after` We need to overwrite certain macros from amsmath. The method `\eqLamsmath@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 `\eqLamsmath@before` will execute argument #1 before any future loading of amsmath. `\eqLamsmath@undefine` undefines a macro in this way and `\eqLamsmath@let` overwrites a macro of `\amsmath/`:

```
104 \def\eqLamsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
105 \def\eqLamsmath@before#1{%
106   \@ifpackageloaded{amsmath}{}{\AddToHook{package/amsmath/before}{#1}}
107 \def\eqLamsmath@undefine#1{\eqLamsmath@before{\let#1\@undefined}}
108 \def\eqLamsmath@let#1#2{\eqLamsmath@undefine#1\let#1#2}
```

**TODO:** temporary fix for development stages

```
109 \@ifpackageloaded{amsmath}{}{
110   \DeclareHookRule{package/amsmath/after}
111   {eqnlines}{after}{latex-lab-testphase-math}}
```

## B.7 PDF Tagging Support

`\eqLtagging@...` 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

```
112 \let\eqLtagging@on\eqLfalse
113 \IfFormatAtLeastTF{2025-06-01}{%
114   \csname tag_if_active:T\endcsname{\let\eqLtagging@on\eqLtrue}}{}
115 \ifdefined\eqLtagging@on
116   \def\eqLtagging@mathsave{%
117     \UseTaggingSocket{math/luamml/save/nNn}{{}\displaystyle{mtd}}
118   \def\eqLtagging@mathaddlast{%
119     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
```

---

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

```

120 \def\eql@tagging@tagbegin{%
121   \UseTaggingSocket{math/display/tag/begin}}
122 \def\eql@tagging@tagend{%
123   \UseTaggingSocket{math/display/tag/end}}
124 \def\eql@tagging@tagsave{%
125   \UseTaggingSocket{math/luamml/mtable/tag/save}}
126 \def\eql@tagging@tagaddbox{%
127   \setbox\z@\copy\eql@tagbox@%
128   \UseTaggingSocket{math/luamml/mtable/tag/set}}
129 \def\eql@tagging@tablesaveinner{%
130   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
131 \def\eql@tagging@tableaddinner{%
132   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
133 \def\eql@tagging@tablesavelines{%
134   \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
135 \def\eql@tagging@tablesavealign{%
136   \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
137 \def\eql@tagging@alignleft{%
138   \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
139 \def\eql@tagging@aligncenter{%
140   \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
141 \def\eql@tagging@alignright{%
142   \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:

```

143 \let\eql@single@doscan\eql@true
144 \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:

```

145 \def\eql@tagging@start{%
146   \csname bool_if:N\expandafter\endcsname
147   \csname l__math_collected_bool\endcsname{%
148     \toks@\expandafter{\eql@tagging@opt}}%
149     \edef\eql@tmp{{\currentenv}{\the\toks@} \the\eql@scan@reg@}}%
150     \csname __math_process:nn\expandafter\endcsname\eql@tmp
151     \@kernel@math@registered@begin
152     \csname bool_set_true:N\expandafter\endcsname
153     \csname l__math_collected_bool\endcsname
154   }%
155 }
156 \def\eql@tagging@end{}
157 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}
158 \else
159 \def\eql@tagging@mathsave{}
160 \def\eql@tagging@mathaddlast{}
161 \def\eql@tagging@tagbegin{}
162 \def\eql@tagging@tagend{}
163 \def\eql@tagging@tagsave{}
164 \def\eql@tagging@tagaddbox{}
165 \def\eql@tagging@tablesaveinner{}
166 \def\eql@tagging@tableaddinner{}

```

```

167 \def\eql@tagging@tablesavelines{}
168 \def\eql@tagging@tablesalign{}
169 \def\eql@tagging@alignleft{}
170 \def\eql@tagging@aligncenter{}
171 \def\eql@tagging@alignright{}
172 \def\eql@tagging@start{}
173 \def\eql@tagging@end{}
174 \def\eql@tagging@register@env{\@gobble}
175 \fi

```

## 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:

```
176 \let\eql@tagsleft\eql@false
```

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

```
177 \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`:

```

178 \def\eql@layoutleftmargin{\leftmargini}
179 \def\eql@layoutleftmarginmax{.5\maxdimen}
180 \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

```

181 \newdimen\eql@tagmargin@
182 \let\eql@tagmargin@val\@undefined
183 \newdimen\eql@tagmargin@ratio@
184 \eql@tagmargin@ratio@\p@
185 \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:

```

186 \newdimen\eql@indent@
187 \def\eql@indent@val{2em}

```

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

```
188 \newdimen\eq@paddingleft@
189 \newdimen\eq@paddingright@
190 \let\eq@paddingleft@val\@undefined
191 \let\eq@paddingright@val\@undefined
```

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

```
\eq@display@marginleft
\eq@display@marginright
192 \let\eq@display@linewidth\@undefined
193 \let\eq@display@marginleft\@undefined
194 \let\eq@display@marginright\@undefined
```

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

```
195 \def\eq@box@colsep{2em}
```

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

```
196 \def\eq@spread@val{\jot}
197 \newdimen\eq@spread@
```

`\eq@tagfuzz@ (dimen)` The value `\eq@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 TeX, nevertheless: **TODO:** probably do not need this due to fixed point arithmetic.

```
198 \newdimen\eq@tagfuzz@
199 \eq@tagfuzz@16sp\relax
```

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

```
200 \def\eq@display@height\@undefined
201 \def\eq@display@depth\@undefined
```

`\eq@skip@mode@short` The setting `\eq@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:

```
202 \def\eq@skip@mode@short{2}

203 \def\eq@skip@mode@cont@above{2}
204 \def\eq@skip@mode@cont@below{0}

205 \def\eq@skip@mode@par@above{3}
206 \def\eq@skip@mode@par@below{0}

207 \def\eq@skip@mode@top@above{4}
208 \def\eq@skip@mode@top@below{0}

209 \newcount\eq@skip@mode@leave@
210 \let\eq@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 211 \newcount\eql@skip@mode@above@
mode@above@ (counter) 212 \newcount\eql@skip@mode@below@
mode@below@ (counter) 213 \let\eql@skip@force@above@\undefined
214 \let\eql@skip@force@below@\undefined
215 \let\eql@skip@custom@above@\undefined
216 \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`:

```
217 \def\eql@skip@long@above{\abovedisplayskip}
218 \def\eql@skip@long@below{\belowdisplayskip}
219 \def\eql@skip@short@above{\abovedisplayshortskip}
220 \def\eql@skip@short@below{\belowdisplayshortskip}
221 \def\eql@skip@cont@above{\eql@skip@short@above}
222 \def\eql@skip@cont@below{\eql@skip@short@below}
223 % \TODO parabove plus parskip?
224 \def\eql@skip@par@above{\eql@skip@long@above}
225 \def\eql@skip@par@below{\eql@skip@long@below}
226 \def\eql@skip@top@above{\eql@skip@long@above}
227 \def\eql@skip@top@below{\eql@skip@long@below}
228 \def\eql@skip@med@above{\abovedisplayskip/2}
229 \def\eql@skip@med@below{\belowdisplayskip/2}
230 \def\eql@skip@tag@above{\z@skip}
231 \def\eql@skip@tag@below{\z@skip}
232 \def\eql@skip@partag@above{\z@skip}
233 \def\eql@skip@partag@below{\z@skip}
234 \def\eql@skip@medtag@above{\z@skip}
235 \def\eql@skip@medtag@below{\z@skip}
```

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

```
236 \newdimen\eql@colsepmin@
237 \def\eql@colsepmin@val{1em}
238 \def\eql@colsepmax@val{.5\maxdimen}
```

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

```
239 \newdimen\eql@tagwidthmin@
240 \eql@tagwidthmin@\z@
```

`\eql@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eql@tagsepmin@`. T<sub>E</sub>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.

---

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



```

241 \newdimen\eql@tagsepmin@
242 \def\eql@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

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

```

\eqlequations@ang@opt
\eql@box@ang@opt
243 \def\eqlequations@sqr@opt{equation,number}
244 \def\eqlequations@ang@opt{align,number}
245 \def\eql@box@ang@opt{align}

```

## Multi-Line Align Mode.

```

246 \let\eql@columns@fulllength\eql@false

```

## C.2 Registers

**TODO:** describe

### General.

`\eql@cellbox@` (*box*) The box `\eql@cellbox@` holds the present alignment component and `\eql@tagbox@` the `\eql@tagbox@` (*box*) tag for the present line. The corresponding dimensions `\eql@cellwidth@` and `\eql@tagwidth@` (*dimen*) hold their widths. `\eql@prevwidth@` (*dimen*) holds the width of the previous alignment component: **TODO:** adjust

```

\eql@tagwidth@ (dimen)
\eql@prevdepth@ (dimen)
\eql@prevgraf@ (counter)
247 \newbox\eql@cellbox@
248 \newbox\eql@tagbox@
249 \newdimen\eql@cellwidth@
250 \newdimen\eql@prevwidth@
251 \newdimen\eql@tagwidth@
252 \newdimen\eql@prevdepth@
253 \newcount\eql@prevgraf@

```

```

\eql@totalwidth@ (dimen)
\eql@tagwidth@max@ (dimen)
254 \newdimen\eql@totalwidth@
255 \newdimen\eql@tagwidth@max@

```

`\eql@line@height@` (*dimen*) The dimension registers `\eql@line@height@` and `\eql@line@depth@` (*dimen*) keep track of the height and depth of the present line in an alignment:

```

256 \newdimen\eql@line@height@
257 \newdimen\eql@line@depth@

```

```

\eql@line@width@ (dimen)
\eql@line@avail@ (dimen)
\eql@line@pos@ (dimen)
\eql@line@widthsep@ (counter)
\eql@line@availsep@ (counter)
\eql@line@possep@ (counter)
\eql@line@offset@ (dimen)
258 \newdimen\eql@line@width@
259 \newdimen\eql@line@avail@
260 \newdimen\eql@line@pos@
261 \newcount\eql@line@availsep@
262 \newcount\eql@line@widthsep@
263 \newcount\eql@line@possep@
264 \newdimen\eql@line@offset@

```

## Rows and Columns.

`\eql@row@` (*counter*) `\eql@row@` counts the present row (1-based) and `\eql@totalrows@` holds the total number of rows:

```
\eql@tagrows@ (counter)
265 \newcount\eql@row@
266 \newcount\eql@totalrows@
267 \newcount\eql@tagrows@
```

```
\eql@column@
\eql@totalcolumns@
268 \newcount\eql@column@
269 \newcount\eql@totalcolumns@
```

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

```
270 \newdimen\eql@colsep@
```

`\eql@columns@inter@` (*counter*)

```
271 \newcount\eql@columns@inter@
```

## Vertical Spacing Adjustments.

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

```
272 \newdimen\eql@display@firstavail@
273 \def\eql@display@firstavail@set#1{%
274   \ifnum\eql@row@=\@ne
275     \global\eql@display@firstavail@#1%
276   \fi
277 }
```

`\eql@firstlast@` (*counter*) The counter stores whether the tag on the first/last line is raised/lowered as 1/2 (or 3 for both). This implies a different `\vskip` corresponding to the mostly empty line:

```
278 \newcount\eql@raisetag@firstlast@
```

## 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:

```
279 \ifdefined\measuring@true\else
280   \expandafter\newif\csname ifmeasuring@\endcsname
281 \fi
282 \AddToHook{package/hyperref/after}{
283   \ifdefined\Hy@ifnotmeasuring
284     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
285   \fi
286 }
```

`\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:

```
287 \ifdefined\@displaytrue\else
288   \expandafter\newif\csname if@display\endcsname
289   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
290 \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  $\text{\LaTeX}$  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

```
291 \let\eql@hook@blockbefore\@empty
292 \let\eql@hook@blockafter\@empty
293 \let\eql@hook@blockin\@empty
294 \let\eql@hook@blockout\@empty
295 \let\eql@hook@linein\@empty
296 \let\eql@hook@lineout\@empty
297 \let\eql@hook@colin\@empty
298 \let\eql@hook@colout\@empty
299 \let\eql@hook@eqin\@empty
300 \let\eql@hook@eqout\@empty
301 \let\eql@hook@innerleft\@empty
302 \let\eql@hook@innerright\@empty
303 \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`

```
304 \let\eql@punct@col\@empty
305 \let\eql@punct@line\relax
306 \let\eql@punct@block\relax
```

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

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

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

```
\eqnpunct
308 \def\eqnpunctcol{%
309   \eql@ifstar@tight\eql@punct@col@setrelax\eql@punct@col@set}
310 \def\eql@punct@col@set#1{\def\eql@punct@col{#1}\ignorespaces}
```

```

311 \def\eql@punct@col@setrelax{\let\eql@punct@col\@empty\ignorespaces}
312 \def\eqnpunctline{%
313   \eql@ifstar@tight\eql@punct@line@setrelax\eql@punct@line@set}
314 \def\eql@punct@line@set#1{\def\eql@punct@line{#1}\ignorespaces}
315 \def\eql@punct@line@setrelax{\let\eql@punct@line\relax\ignorespaces}
316 \def\eqnpunctdefault{%
317   \eql@ifstar@tight\eql@punct@main@setrelax\eql@punct@main@set}
318 \def\eql@punct@main@set#1{\def\eql@punct@main{#1}\ignorespaces}
319 \def\eql@punct@main@setrelax{\let\eql@punct@main\relax\ignorespaces}
320 \def\eqnpunct{%
321   \eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
322 \def\eql@punct@next@set#1{%
323   \ifmmode
324     \def\eql@punct@col{#1}%
325     \def\eql@punct@line{#1}%
326     \def\eql@punct@block{#1}%
327   \else
328     \eqnaddopt{punct={#1}}%
329   \fi
330   \ignorespaces}
331 \def\eql@punct@next@setrelax{%
332   \ifmmode
333     \let\eql@punct@block\relax
334   \else
335     \eqnaddopt{punct*}%
336   \fi
337   \ignorespaces}

```

`\eql@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:

```

338 \def\eql@punct@apply@col{%
339   \ifx\eql@punct@col\@empty\else
340     \eql@punct@sep
341     \eql@punct@col
342     \let\eql@punct@col\@empty
343   \fi
344 }

```

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

`\eql@punct@apply@line`

```

345 \def\eql@punct@apply@line{%
346   \ifx\eql@punct@line\relax
347 % \TODO hand down immediately?
348   \else
349     \ifx\eql@punct@line\@empty\else
350       \eql@punct@sep
351       \eql@punct@line
352     \fi
353     \let\eql@punct@line\relax
354     \let\eql@punct@col\@empty
355   \fi
356 }

```

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

```

357 \def\eql@punct@apply@block{%
358   \ifx\eql@punct@block\relax
359   % \TODO hand down immediately?
360   \else
361     \ifx\eql@punct@block\@empty\else
362       \eql@punct@sep
363       \eql@punct@block
364     \fi
365     \let\eql@punct@block\relax
366     \let\eql@punct@line\relax
367     \let\eql@punct@col\@empty
368   \fi
369 }
370 \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:

```

371 \def\eql@class@innerright@sel{%
372   \ifdim\eql@prevwidth@=\z@
373     \eql@class@innerlead
374   \else
375     \eql@class@innerright
376   \fi
377 }

```

`\@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`

```

378 \def\eql@class@innerleft@set#1{%
379   \def\eql@class@innerleft{#1}%
380 }
381 \def\eql@class@innerright@set#1{%
382   \def\eql@class@innerright{#1}%
383   \let\eql@class@innerright@sel\eql@class@innerright
384 }
385 \def\eql@class@innerlead@set#1{%
386   \def\eql@class@innerlead{#1}%
387   \let\eql@class@innerright@sel\eql@class@innerright@sel@
388 }

```

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

```

389 \def\eql@class@ampeq{%
390   \eql@class@innerleft@set{}%
391   \eql@class@innerright@set{}}%
392 }
393 \def\eql@class@eqamp{%
394   \eql@class@innerleft@set{\mathrel{}}%
395   \eql@class@innerright@set{\mathrel{}}%

```

```

396 \eq@class@innerlead@set{}}}%
397 }
398 \eq@class@ampeq

```

### D.3 Framed Cells

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

```

399 \let\eq@frame@cmd\@undefined
400 \newdimen\eq@frame@margin@
401 \def\eq@frame@set[#1]{\global\def\eq@frame@cmd{#1}}
402 \def\eq@frame@reset{\global\let\eq@frame@cmd\@undefined}
403 \protected\def\framecell{\eq@testopt@tight@ampsafe\eq@frame@set\fbx}
404 \def\eq@frame@measure{%
405   \setbox\z@\hbox{\eq@frame@cmd{}}}%
406   \eq@frame@margin@.5\wd\z@
407 }
408 \def\eq@frame@print{%
409   \setbox\eq@cellbox@\hbox{%
410     \eq@frame@cmd{\unhbox\eq@cellbox@}%
411   }%
412   \eq@frame@reset
413 }
414 \def\eq@frame@adjust{%
415   \setbox\eq@cellbox@\hbox{%
416     \eq@frame@cmd{%
417       \unhbox\eq@cellbox@
418       \unkern
419       \unskip
420     }%
421     \hfil
422     \kern\z@
423   }%
424   \eq@frame@reset
425 }

```

### D.4 Alternative Content Description

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

```

426 \DeclareRobustCommand{\eqnalt}[2][{}]{

```

## E Equation Numbering

**TODO:** describe

### E.1 Supporting Definitions

Parameters.

```

427 \let\eq@numbering@autolabel\eq@false
428 \let\eq@numbering@autotag\eq@true
429 \let\eq@numbering@warn\eq@true

```

```
430 \def\eql@labelname@generic{[equation]}
```

## Registers.

```
431 \let\eql@numbering@mode\@undefined
432 \let\eql@numbering@measure@call\@empty
```

```
ring@target@ (counter)
```

```
433 \let\eql@numbering@active\eql@true
434 \newcount\eql@numbering@target@
```

```
435 \let\eql@nextlabel\@undefined
436 \let\eql@nextlabelname\@undefined
437 \let\eql@blocklabel\@undefined
438 \let\eql@blocklabelname\@undefined
```

```
\eql@nexttag
```

```
439 \let\eql@nexttag\@undefined
440 \let\eql@blocktag\@undefined
```

```
setag@amount@ (dimen)
```

```
441 \newdimen\eql@raisetag@amount@
```

```
g@refcount@ (counter)
```

```
442 \newcount\eql@numbering@refcount@
443 \eql@numbering@refcount@z@
444 \def\eql@numbering@ref{equation.eql-\the\eql@numbering@refcount@}
445 \def\eql@numbering@refstep{\global\advance\eql@numbering@refcount@\@ne}
```

## E.2 Schemes

**TODO:** describe

### Table.

```
446 \def\eql@numbering@tab@first{first}
447 \def\eql@numbering@tab@last{last}
448 \def\eql@numbering@tab@middle{middle}
449 \def\eql@numbering@tab@here{here}
450 \def\eql@numbering@tab@best{best}
451 \def\eql@numbering@tab@in{in}
452 \def\eql@numbering@tab@out{out}
453 \def\eql@numbering@tab@sub{sub}
454 \def\eql@numbering@tab@all{all}
455 \def\eql@numbering@tab@none{none}
```

**TODO:** describe

```
456 \let\eql@numbering@tab@f\eql@numbering@tab@first
457 \let\eql@numbering@tab@l\eql@numbering@tab@last
458 \let\eql@numbering@tab@m\eql@numbering@tab@middle
459 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
460 \let\eql@numbering@tab@o\eql@numbering@tab@out
```

```

461 \let\eql@numbering@tab@outside\eql@numbering@tab@out
462 \let\eql@numbering@tab@i\eql@numbering@tab@in
463 \let\eql@numbering@tab@inside\eql@numbering@tab@in
464 \let\eql@numbering@tab@within\eql@numbering@tab@in
465 \let\eql@numbering@tab@h\eql@numbering@tab@here
466 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
467 \let\eql@numbering@tab@s\eql@numbering@tab@sub
468 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
469 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
470 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
471 \let\eql@numbering@tab@a\eql@numbering@tab@all
472 \let\eql@numbering@tab@n\eql@numbering@tab@none
473 \expandafter\let\csname eql@numbering@tab@!\endcsname\eql@numbering@tab@all
474 \expandafter\let\csname eql@numbering@tab@*\endcsname\eql@numbering@tab@none
475 \expandafter\let\csname eql@numbering@tab@1\endcsname\eql@numbering@tab@first
476 \expandafter\let\csname eql@numbering@tab@+\endcsname\eql@numbering@tab@best
477 \let\eql@numbering@mode\eql@numbering@tab@all

```

### Implementations.

```

478 \def\eql@numbering@mode@all{%
479   \eql@numbering@target@{m@ne}
480 \def\eql@numbering@mode@first{%
481   \eql@numbering@target@{ne}
482 \def\eql@numbering@mode@last{%
483   \eql@numbering@target@{MM}
484 \def\eql@numbering@mode@here{%
485   \eql@numbering@target@{z@}

```

**TODO:** describe

```

486 \def\eql@numbering@mode@in{%
487   \ifdefined\eql@tagsleft
488     \eql@numbering@mode@last
489   \else
490     \eql@numbering@mode@first
491   \fi
492 }

```

**TODO:** describe

```

493 \def\eql@numbering@mode@out{%
494   \ifdefined\eql@tagsleft
495     \eql@numbering@mode@first
496   \else
497     \eql@numbering@mode@last
498   \fi
499 }

```

**TODO:** describe

```

500 \def\eql@numbering@mode@middle{%
501   \eql@numbering@target@{z@}
502   \let\eql@numbering@measure@call\eql@numbering@measure@middle}
503 \def\eql@numbering@measure@middle{%
504   \ifnum\eql@numbering@target@={z@}
505     \eql@numbering@target@=\eql@totalrows@
506     \advance\eql@numbering@target@{one}
507     \divide\eql@numbering@target@{two}
508   \fi

```



509 }

**TODO:** describe

```
510 \def\eq@numbering@mode@best{%
511   \eq@numbering@target@z@
512   \let\eq@numbering@measure@call\eq@numbering@measure@best
513 }
514 \def\eq@numbering@measure@best{%
515   \ifnum\eq@numbering@target@z@
516     \let\eq@numbering@best@use\eq@true
517     \eq@numbering@mode@out
518   \fi
519 }
```

**TODO:** describe

```
520 \def\eq@numbering@mode@sub{%
521   \eq@numbering@target@m@ne
522   \ifdefined\eq@subequations@active\else
523     \let\eq@numbering@measure@call\eq@numbering@subeq@test
524     \let\eq@numbering@subeq@use\eq@true
525   \fi
526 }
```

**Selection.**

```
527 \def\eq@numbering@set#1{%
528   \ifcsname eq@numbering@tab@#1\endcsname
529     \expandafter\let\expandafter\eq@numbering@mode
530       \csname eq@numbering@tab@#1\endcsname
531     \ifx\eq@numbering@mode\eq@numbering@tab@none
532       \let\eq@numbering@mode\eq@numbering@tab@all
533       \let\eq@numbering@active\eq@false
534     \fi
535   \else
536     \eq@error{numbering mode ‘#1’ unknown: setting to ‘all’}%
537     \let\eq@numbering@mode\eq@numbering@tab@all
538   \fi
539 }
```

**TODO:** describe

```
540 \def\eq@numbering@mode@eval{%
541   \let\eq@numbering@measure@call\@empty
542   \let\eq@numbering@subeq@use\eq@false
543   \csname eq@numbering@mode@\eq@numbering@mode\endcsname
544   \ifdefined\eq@numbering@active
545     \let\eq@numbering@eqnswinit\@eqnswtrue
546   \else
547     \let\eq@numbering@eqnswinit\@eqnswfalse
548   \fi
549   \let\eq@numbering@active\eq@false
550 }
```

## E.3 Interface

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

```

551 \eql@amsmath@after{
552   \let\eql@print@eqnum@default\print@eqnum
553   \let\eql@incr@eqnum@default\incr@eqnum
554 }

```

**TODO:** describe

```

555 \protected\def\donumber{%
556   \if@eqnsw\else
557     \global\@eqnswtrue
558     \ifx\print@eqn\@empty
559       \global\let\print@eqn\eql@print@eqnum@default
560     \fi
561     \ifx\incr@eqn\@empty
562       \global\let\incr@eqn\eql@incr@eqnum@default
563     \fi
564   \fi
565 }

```

**TODO:** reconsider operation

\numberhere

```

566 \protected\def\eql@numberhere{%
567   \global\eql@numbering@target@=\eql@row@
568 }

```

**TODO:** describe

\numbernext

```

569 \protected\def\eql@numbernext{%
570   \ifnum\eql@numbering@target@=\eql@row@
571     \global\advance\eql@numbering@target@\@ne
572   \fi
573 }

```

**Activation Trigger.**

```

574 \def\eql@numbering@autoenable{%
575   \global\@eqnswtrue
576   \ifx\eql@numbering@mode\eql@numbering@tab@here
577     \ifnum\eql@numbering@target@=\z@
578       \numberhere
579     \fi
580   \fi
581 }

```

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

\eql@label@org

```

582 \let\eql@label@org\label

```

**TODO:** describe

```

583 \def\eql@label@gobble{\eql@ampprotect\eql@testopt@tight\eql@gobbleoptone{}}

```

**TODO:** describe

```

584 \protected\def\eql@label{%
585   \ifdefined\eql@numbering@autolabel
586     \eql@numbering@autoenable
587   \fi
588   \eql@ampprotecttwo{\eql@ifnextchar@tight[]\eql@label@name\eql@label@plain%]
589 }

```

**TODO:** describe

```

590 \def\eql@label@plain#1{%
591   \ifdefined\eql@numbering@warn
592     \ifdefined\eql@nextlabel
593       \eql@warn@label@multiple\eql@nextlabel
594     \fi
595   \fi
596   \global\edef\eql@nextlabel{#1}%
597 }

```

**TODO:** describe

```

598 \def\eql@label@name[#1]#2{%
599   \ifdefined\eql@numbering@warn
600     \ifdefined\eql@nextlabel
601       \eql@warn@label@multiple\eql@nextlabel{#2}%
602     \fi
603     \ifdefined\eql@nextlabelname
604       \eql@warn@labelname@multiple
605     \fi
606   \fi
607   \protected@edef\eql@nextlabelname{#1}%
608   \global\let\eql@nextlabelname\eql@nextlabelname
609   \global\edef\eql@nextlabel{#2}%
610 }

```

**TODO:** describe

```

611 \def\eql@blocklabel@set#1{%
612   \ifdefined\eql@blocklabel
613     \eql@warn@label@multiple\eql@blocklabel{#1}%
614   \fi
615   \edef\eql@blocklabel{#1}%
616 }

```

**TODO:** describe

```

617 \def\eql@blocklabelname@set#1{%
618   \ifdefined\eql@blocklabelname
619     \eql@warn@labelname@multiple
620   \fi
621   \protected@edef\eql@blocklabelname{#1}%
622 }

```

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

`\eql@tag@default`

```

623 \protected\def\eql@tag@default{%
624   \eql@error{\string\tag\space not allowed here}{ }\eql@tag@gobble}
625 \let\tag\eql@tag@default

```

`\eqL@tag@gobble`

```
626 \def\eqL@tag@gobble{%
627   \eqL@ampprotecttwo\eqL@teststaropt@tight\eqL@gobbleoptone\eqL@gobbleoptone{}}
```

**TODO:** describe

```
628 \protected\def\eqL@tag{%
629   \ifdefined\eqL@numbering@autotag
630     \eqL@numbering@autoenable
631   \fi
632   \ifdefined\eqL@numbering@warn
633     \ifdefined\eqL@nexttag
634       \eqL@warn@tag@multiple
635     \fi
636   \fi
637   \eqL@ampprotecttwo\eqL@teststaropt@tight
638   {\eqL@nexttag@set\z@}{\eqL@nexttag@set\@ne}\eqL@tmp
639 }
```

`\eqL@nexttag@set`

```
640 \def\eqL@nexttag@set#1[#2]#3{%
641   \protected@edef\eqL@tmp{#3}%
642   \protected@edef\eqL@nexttag{{#1}{#2}}{\eqL@tmp}}%
643   \global\let\eqL@nexttag\eqL@nexttag
644 }
```

`\eqL@tag@eval`

```
645 \def\eqL@tag@eval#1#2#3{%
646   \def\eqL@tag@label{#2}%
647   \def\eqL@tag@text{#3}%
648   \ifnum#1=\@ne
649     \let\eqL@tag@tool\eqL@tag@form
650     \protected@edef\eqL@tag@label{\p@equation\eqL@tag@label}%
651   \else
652     \let\eqL@tag@tool\@firstofone
653   \fi
654 }
```

**TODO:** describe

```
655 \def\eqL@blocktag@set#1#2{%
656   \ifdefined\eqL@blocktag
657     \eqL@warn@tag@multiple
658   \fi
659   \protected@edef\eqL@blocktag{{#1}{#2}{#2}}%
660 }
```

**Raise Tags.**

`\raisetag`

```
661 \def\eqL@raisetag@default{%
662   \eqL@warning{\string\raisetag\space not allowed here}
663   \@gobble
664 }
```

**TODO:** describe

```
665 \eq\amsmath\let\raisetag\eq\raisetag@default
```

**TODO:** maybe introduce a star form to enforce raise?

```
666 \def\eq\raisetag#1{\global\eq\raisetag@amount@\glueexpr#1\relax}
```

```
667 \let\eq\raisetag@gobble\@gobble
```

## E.4 Integration

**TODO:** describe

**Single Line.** **TODO:** describe

```
668 \def\eq\numbering@single@init{%
669   \let\eq\numbering@warn\eq\true
670   \let\label\eq\label
671   \let\tag\eq\tag
672   \let\raisetag\eq\raisetag
673   \let\numberhere\donumber
674   \let\numbernext\nonumber
675   \eq\numbering@target@\m@ne
676   \let\eq\nextlabel\eq\blocklabel
677   \let\eq\nextlabelname\eq\blocklabelname
678   \let\eq\nexttag\eq\blocktag
679   \eq\numbering@eqnswinit
680   \ifdefined\eq\numbering@autolabel
681     \ifdefined\eq\nextlabel
682       \eqnswtrue
683     \fi
684   \fi
685   \ifdefined\eq\numbering@autotag
686     \ifdefined\eq\nexttag
687       \eqnswtrue
688     \fi
689   \fi
690   \global\eq\raisetag@amount@\z@
691 }
```

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

```
692 \def\eq\numbering@measure@init{%
693   \let\eq\numbering@warn\eq\true
694   \let\label\eq\label
695   \let\tag\eq\tag
696   \let\raisetag\eq\raisetag
697   \ifnum\eq\numbering@target@<\z@
698     \let\numberhere\donumber
699     \let\numbernext\nonumber
700   \else
701     \let\numberhere\eq\numberhere
702     \let\numbernext\eq\numbernext
703     \eq\numbering@eqnswinit
704     \ifdefined\eq\numbering@autolabel
705       \ifdefined\eq\nextlabel
706         \eqnswtrue
707     \fi
```

```

708   \fi
709 \fi
710 \let\eql@nextlabel\eql@blocklabel
711 \let\eql@nextlabelname\eql@blocklabelname
712 \let\eql@nexttag\eql@blocktag
713 }

```

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

```

714 \def\eql@numbering@measure@line@begin{%
715   \ifnum\eql@numbering@target@<\z@
716     \global\eql@numbering@eqnswinit
717   \fi
718 }

```

**TODO:** describe

```

719 \def\eql@numbering@measure@eval{%
720   \let\eql@numbering@best@use\eql@false
721   \eql@numbering@measure@call
722   \ifnum\eql@numbering@target@>\eql@totalrows@
723     \eql@numbering@target@\eql@totalrows@
724   \fi
725   \ifnum\eql@numbering@target@>\z@
726     \if@eqnsw\else
727       \eql@numbering@target@\z@
728     \fi
729   \fi
730   \ifnum\eql@numbering@target@<\@ne
731     \ifdefined\eql@nextlabel
732       \eql@warn@label@unused
733       \let\eql@nextlabel\@undefined
734     \fi
735     \ifdefined\eql@nexttag
736       \eql@warn@tag@unused
737       \let\eql@nexttag\@undefined
738     \fi
739     \ifdefined\eql@nextlabelname
740       \eql@warn@labelname@unused
741       \let\eql@nextlabelname\@undefined
742     \fi
743   \fi
744 }

```

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

```

745 \def\eql@numbering@print@init{%
746   \ifnum\eql@numbering@target@<\z@
747     \let\eql@numbering@warn\eql@false
748     \let\label\eql@label
749     \let\tag\eql@tag
750     \let\raisetag\eql@raisetag
751     \let\numberhere\donumber
752     \let\numbernext\nonumber
753     \let\eql@nextlabel\eql@blocklabel
754     \let\eql@nextlabelname\eql@blocklabelname
755     \let\eql@nexttag\eql@blocktag
756   \else
757     \let\label\eql@label@gobble

```

```

758 \let\tag\eq\tag@gobble
759 \let\raisetag\eq\raisetag@gobble
760 \let\numberhere\@empty
761 \let\numbernext\@empty
762 \fi
763 }

```

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

```

764 \def\eq\@numbering@print@block@begin{%
765 \ifnum\eq\@numbering@target@>\z@
766 \eq\@numbering@makeblockanchor
767 \fi
768 \ifdefined\eq\@numbering@subeq@use
769 \eq\@numbering@printsubeqlabel
770 \fi
771 }

```

**TODO:** describe

```

772 \def\eq\@numbering@print@line@begin{%
773 \ifnum\eq\@numbering@target@<\z@
774 \global\eq\@numbering@eqnswinit
775 \global\eq\@raisetag@amount@<\z@
776 \fi
777 }

```

**TODO:** describe

```

778 \def\eq\@numbering@print@line@eval{%
779 \ifnum\eq\@numbering@target@<\z@ \else
780 \ifnum\eq\@numbering@target@=\eq\@row@
781 \global\@eqnswtrue
782 \else
783 \global\@eqnswfalse
784 \fi
785 \fi
786 }

```

## E.5 Component Display

**Showkeys Integration.** **TODO:** describe

```

787 \let\eq\SK@loaded\eq@false
788 \let\eq\SK@label\@gobble
789 \let\eq\SK@clearlabel\@empty
790 \let\eq\SK@setlabel\@gobble
791 \let\eq\SK@printlabel@right\@empty
792 \let\eq\SK@printlabel@left\@empty
793 \let\eq\SK@printlabel@line\@empty
794 \def\eq\@label@clean{\eq\@label@org}
795 \AddToHook{package/showkeys/after}{
796 \let\eq\SK@loaded\eq@true
797 \def\eq\SK@label#1{\SK@\SK@@label#1}
798 \def\eq\SK@clearlabel{\let\eq\SK@lab\relax}
799 \eq\SK@clearlabel
800 \def\eq\SK@@label#1>#2\SK@{%
801 \def\eq\SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
802 }

```

```

803 \def\eq@SK@setlabel#1{\SK@\eq@SK@@label#1}
804 \def\eq@SK@printlabel@right{%
805   \ifx\eq@SK@lab\relax\else
806     \rlap{\kern\marginparsep\eq@SK@lab}%
807     \eq@SK@clearlabel
808   \fi
809 }
810 \def\eq@SK@printlabel@left{%
811   \ifx\eq@SK@lab\relax\else
812     \llap{\eq@SK@lab\kern\marginparsep}%
813     \eq@SK@clearlabel
814   \fi
815 }
816 \def\eq@SK@printlabel@line{%
817   \ifx\eq@SK@lab\relax\else
818     \dimen@ \prevdepth
819     \nointerlineskip
820     \ifdefined\eq@tagsleft
821       \llap{%
822         \eq@SK@lab
823         \kern\marginparsep
824       }%
825       \eq@SK@clearlabel
826     \else
827       \rlap{%
828         \dimen@ \displaywidth
829         \advance\dimen@ \marginparsep
830         \kern\dimen@
831         \eq@SK@lab
832       }%
833     \fi
834     \eq@SK@clearlabel
835     \prevdepth \dimen@
836   \fi
837 }
838 \let\eq@label@org\label
839 \def\eq@label@clean{\let\SK@\@gobbletwo\eq@label@org}
840 }

```

## Labels.

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

```

841 \def\eq@compose@label{%
842   \eq@SK@clearlabel
843   \ifdefined\eq@nextlabel
844     \ifnum
845       \ifnum\eq@numbering@target@<\z@
846         \eq@row@
847       \else
848         \eq@numbering@target@
849       \fi=\eq@row@
850     \eq@compose@label@
851   \fi
852 \fi
853 }

```



`\eql@compose@label@` **TODO:** describe

```
854 \def\eql@compose@label@{%
855   \eql@SK@setlabel\eql@nextlabel
856   \begingroup
857   \ifnum\eql@numbering@target@=\eql@row@
858     \eql@numbering@setblockanchor
859   \fi
860   \ifdefined\eql@nextlabelname
861     \let\@currentlabelname\eql@nextlabelname
862     \let\eql@nextlabelname\@undefined
863   \else
864     \let\@currentlabelname\eql@labelname@generic
865   \fi
866   \expandafter\eql@label@clean\expandafter{\eql@nextlabel}%
867   \global\let\eql@nextlabel\@undefined
868 \endgroup
869 }
```

**TODO:** describe

```
870 \def\eql@numbering@printsubeqlabel{%
871   \ifdefined\eql@parentlabel
872     \eql@numbering@makeblockanchor
873     \eql@SK@setlabel\eql@parentlabel
874     \begingroup
875     \def\@currentcounter{equation}%
876     \eql@numbering@setblockanchor
877     \let\@currentlabelname\eql@labelname@generic
878     \protected@edef\@currentlabel{\p@equation\theparentequation}%
879     \expandafter\eql@label@clean\expandafter{\eql@parentlabel}%
880   \endgroup
881   \eql@SK@printlabel@line
882 \fi
883 }
```

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

```
884 \let\eql@Hy@anchor\@gobble
885 \AddToHook{package/hyperref/after}{
886   \def\eql@Hy@anchor#1{%
887     \Hy@raisedlink{\hyper@anchor{#1}}%
888   }%
889 }
```

**TODO:** describe

```
890 \def\eql@numbering@makeblockanchor{%
891   \eql@numbering@refstep
892   \global\edef\eql@label@currentHref{\eql@numbering@ref}%
893   \eql@Hy@anchor\eql@label@currentHref
894   \global\edef\eql@label@thepage{\thepage}%
895 }
896 \def\eql@numbering@setblockanchor{%
897   \let\thepage\eql@label@thepage
898   \let\@currentHref\eql@label@currentHref
899 }
```

**TODO:** describe

`\eql@compose@anchor`

```
900 \def\eql@compose@anchor{%
901   \ifdefined\eql@nexttag
902     \expandafter\eql@tag@eval\eql@nexttag
903     \def\@currentcounter{equation}%
904     \let\@currentlabel\eql@tag@label
905     \eql@numbering@refstep
906     \edef\@currentHref{\eql@numbering@ref}%
907     \eql@Hy@anchor\@currentHref
908     \global\let\eql@nexttag\@undefined
909   \else
910     \refstepcounter{equation}%
911     \let\eql@tag@tool\eql@tag@form
912     \edef\eql@tag@text{\theequation}%
913   \fi
914 }
```

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

```
915 \def\eql@tag@setbox@#1{%
916   \def\eql@tag@box##1{#1}%
917 }
918 \def\eql@tag@setbox#1{%
919   \def\eql@tag@box##1{\hbox{\m@th\normalfont#1}}%
920 }
```

**TODO:** describe

```
921 \def\eql@tag@setform@#1{%
922   \def\eql@tag@form##1{#1}%
923 }
924 \def\eql@tag@setform#1#2#3{%
925   \def\eql@tag@form##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
926 }

927 \eql@tag@setbox{#1}
928 \eql@tag@setform({#1})
929 \def\eql@tag@boxedform#1{\eql@tag@box{\eql@tag@form{#1}}}%

930 \protected\def\tagform{\eql@tag@form}
931 \protected\def\tagbox{\eql@tag@box}
932 \protected\def\tagboxed{\eql@tag@boxedform}
```

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

```
933 \protected\def\eqref#1{\textup{\eql@tag@boxedform{\ref{#1}}}}
```

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

```
934 \def\eql@compose@tag{%
935   \eql@tagging@tagbegin
936   \eql@tag@box{%
937     \eql@tag@tool\eql@tag@text
938     \eql@tagging@tagsave
939   }%
940   \eql@tagging@tagend
941 }
```

## E.6 Tag Composition

**TODO:** describe

```
942 \def\eql@compose@measure{%
943   \ifdefined\eql@nexttag
944     \expandafter\eql@tag@eval\eql@nexttag
945     \eql@tag@box{\eql@tag@tool\eql@tag@text}%
946   \else
947     \stepcounter{equation}%
948     \eql@tag@box{\eql@tag@form\theequation}%
949   \fi
950   \ifnum\eql@numbering@target@<\z@
951     \global\let\eql@nextlabel\@undefined
952     \global\let\eql@nexttag\@undefined
953   \fi
954 }
```

**TODO:** describe

```
955 \def\eql@compose@print{%
956   \eql@compose@anchor
957   \eql@compose@label
958   \ifdefined\eql@tagsleft
959     \eql@SK@printlabel@left
960     \eql@compose@tag
961   \else
962     \eql@compose@tag
963     \eql@SK@printlabel@right
964   \fi
965 }
```

**TODO:** describe

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

```
966 \def\eql@tagbox@make#1{%
967   \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
968   \eql@tagwidth@\wd\eql@tagbox@
969   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
970     \eql@tagwidth@\eql@tagwidthmin@
971   \fi
972   \advance\eql@tagwidth@\eql@tagsepmin@
973 }
```

**TODO:** describe

```
974 \def\eql@tagbox@print@tagsright{%
975   \kern-\wd\eql@tagbox@
976   \box\eql@tagbox@
977 }
```

**TODO:** describe

```
978 \def\eql@tagbox@print@tagsleft{%
979   \wd\eql@tagbox@\z@
980   \box\eql@tagbox@
981 }
```

**TODO:** describe

```
982 \def\eql@tagbox@print@tagsright@raise{%
```

```

983 \ifnum\eq@row@=\eq@totalrows@
984   \global\advance\eq@raisetag@firstlast@\tw@
985 \fi
986 \llap{\vtop{%
987   \vskip-\eq@raisetag@amount@
988   \normalbaselines
989   \setbox\@ne\hbox{}}%
990   \dp\@ne\eq@line@depth@
991   \box\@ne
992   \box\eq@tagbox@
993 }}%
994 }

```

**TODO:** describe

```

995 \def\eq@tagbox@print@tagsleft@raise{%
996   \ifnum\eq@row@=\@ne
997     \global\advance\eq@raisetag@firstlast@\@ne
998   \fi
999   \rlap{\vbox{%
1000     \normalbaselines
1001     \box\eq@tagbox@
1002     \vbox to\eq@line@height@{}}%
1003     \vskip\eq@raisetag@amount@
1004   }}%
1005 }

```

`\eq@tagbox@print@cell`

```

1006 \def\eq@tagbox@print@cell{%
1007   \advance\eq@tagwidth@-\eq@tagfuzz@
1008   \ifdefined\eq@tagsleft
1009     \eq@display@firstavail@set\z@
1010     \ifdim\eq@tagwidth@>\eq@line@avail@
1011       \eq@tagbox@print@tagsleft@raise
1012     \else
1013       \eq@tagbox@print@tagsleft
1014     \fi
1015     \kern\displaywidth
1016   \else
1017     \kern\displaywidth
1018     \advance\eq@tagwidth@\eq@line@width@
1019     \ifdim\eq@tagwidth@>\displaywidth
1020       \eq@tagbox@print@tagsright@raise
1021     \else
1022       \eq@tagbox@print@tagsright
1023     \fi
1024   \fi
1025 }

```

## 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

<code>\parentequation (counter)</code>	We define a counter to store the main equation number while in subequation mode. It makes sense to share this definition with <code>amsmath</code> as <code>parentequation</code> , and we need to undefine it when <code>amsmath</code> is loaded at a later stage:
	<pre> 1026 \eq\amsmath@undefine\c@parentequation 1027 \eq\amsmath@undefine\theparentequation 1028 \ifdefined\c@parentequation\else 1029 \newcounter{parentequation} 1030 \fi </pre>
<code>\subequations@template</code>	We store a template which will be installed as <code>\theequation</code> in subequations mode: <b>TODO:</b> need to protect something?!
	<pre> 1031 \def\eq\subequations@template{\theparentequation\alph{equation}} </pre>
<code>\subequations@active</code>	A boolean register which tells whether subequations are in use and thus must not be invoked again:
	<pre> 1032 \let\eq\subequations@active\eq&gt;false </pre>
<code>\eq\subequations@init</code>	Low-level initialise the subequations mode. Store the equation counter in <code>\eq\subequations@restorecounter</code> for the case that no equation numbers will be used. Step the equation counter, copy it to <code>parentequation</code> and initialise <code>\theparentequation</code> (and its <code>hyperref</code> counterpart) with the expanded current value of <code>\theequation</code> ; fill with tag data instead if a tag has been specified. Reset the equation counter and use the template for <code>\theequation</code> :
	<pre> 1033 \def\eq\subequations@init{% 1034   \edef\eq\subequations@restorecounter{% 1035     \global\c@equation\the\c@equation\relax}% 1036   \ifdefined\eq\blocktag 1037     \expandafter\eq\tag@eval\eq\blocktag 1038     \eq\numbering@refstep 1039     \protected@edef\theHparentequation{\eq\numbering@ref}% 1040     \protected@edef\theparentequation{\eq\tag@text}% 1041   \else 1042     \advance\c@equation\@ne 1043     \protected@edef\theparentequation{\theequation}% 1044     \ifdefined\theHequation 1045       \protected@edef\theHparentequation{\theHequation}% 1046     \fi 1047   \fi 1048   \global\c@parentequation\c@equation 1049   \global\c@equation\z@ 1050   \let\theequation\eq\subequations@template 1051   \def\theHequation{\theHparentequation.\arabic{equation}}}% 1052 } </pre>
<code>\subequations@close</code>	Low-level close the subequations mode. If no number has been used, return to the original equation counter, otherwise use the value stored in <code>parentequation</code> . Note that we cannot use <code>\setcounter</code> here because the <code>calc</code> version would involve actions which are not allowed after <code>\halign</code> within a display equation:
	<pre> 1053 \def\eq\subequations@close{% 1054   \ifnum\c@equation=\z@ 1055     \eq\subequations@restorecounter </pre>

```

1056 \else
1057   \global\c@equation\c@parentequation
1058 \fi
1059 }

```

## F.2 Environment

`\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 `\eql@numbering@printsubeqlabel`

```

1060 \def\eql@subequations@start{%
1061   \let\eql@blocklabel\@undefined
1062   \let\eql@blocklabelname\@undefined
1063   \let\eql@blocktag\@undefined
1064   \eql@nextopt@process{subequations}%
1065   \eql@subequations@init
1066   \eql@numbering@refstep
1067   \edef\eql@subequations@currentHref{\eql@numbering@ref}%
1068   \eql@Hy@anchor\eql@subequations@currentHref
1069   \edef\eql@subequations@thepage{\thepage}%
1070   \def\@currentcounter{equation}%
1071   \let\@currentHref\eql@subequations@currentHref
1072   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1073   \ifdefined\eql@blocklabelname
1074     \let\@currentlabelname\eql@blocklabelname
1075   \else
1076     \let\@currentlabelname\eql@labelname@generic
1077   \fi
1078   \let\eql@subequations@active\eql@true
1079   \ifdefined\eql@blocklabel
1080     \eql@SK@label\eql@blocklabel
1081   \fi
1082   \ignorespaces
1083 }

```

`\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:

```

1084 \def\eql@subequations@end{%
1085   \ifnum\c@equation>\z@
1086     \ifdefined\eql@blocklabel
1087       \begingroup
1088         \def\@currentcounter{equation}%
1089         \let\thepage\eql@subequations@thepage
1090         \let\@currentHref\eql@subequations@currentHref
1091 % \TODO how about tag* ?! also within equations!
1092         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1093         \ifdefined\eql@blocklabelname
1094           \let\@currentlabelname\eql@blocklabelname
1095         \else
1096           \let\@currentlabelname\eql@labelname@generic
1097         \fi
1098         \expandafter\eql@label@clean\expandafter{\eql@blocklabel}%
1099       \endgroup
1100     \fi
1101   \fi
1102   \eql@subequations@close

```

```

1103 \ignorespacesafterend
1104 }

```

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

```

1105 \newenvironment{eq1@subequations}{%
1106 (dev)\eq1@dev@enterenv
1107 \eq1@subequations@testall\eq1@subequations@start%
1108 }{%
1109 \eq1@subequations@end
1110 (dev)\eq1@dev@leaveenv
1111 }

```

**TODO:** describe

```

1112 \def\eq1@subequations@testall{\eq1@subequations@testopt}
1113 \def\eq1@subequations@testopt#1{%
1114 \eq1@ifnextchar@tight[%]
1115 {\eq1@subequations@adopt{\eq1@subequations@testat{#1}}}%
1116 {\eq1@subequations@testat{#1}}}
1117 \def\eq1@subequations@adopt#1[#2]{\eqnadopt{#2}#1}
1118 \def\eq1@subequations@testat#1{%
1119 \eq1@ifat@tight%
1120 {\eq1@subequations@addlabel{#1}}%
1121 {#1}}
1122 \def\eq1@subequations@addlabel#1#2{\eqnadopt{label={#2}}#1}

```

### F.3 Subequation Scheme

**TODO:** describe

```

1123 \def\eq1@numbering@subeq@init{%
1124 \let\eq1@save@theequation\theequation
1125 \let\eq1@save@theHequation\theHequation
1126 \eq1@subequations@init
1127 \let\eq1@parentlabel\eq1@blocklabel
1128 \let\eq1@parentlabelname\eq1@blocklabelname
1129 \let\eq1@parenttag\eq1@blocktag
1130 \let\eq1@blocklabel\@undefined
1131 \let\eq1@blocklabelname\@undefined
1132 \let\eq1@blocktag\@undefined
1133 }

```

**TODO:** describe

```

1134 \def\eq1@numbering@subeq@test{%
1135 \ifnum\c@equation<\tw@
1136 \let\eq1@numbering@subeq@use\@ne
1137 \fi
1138 }

```

**TODO:** describe

```

1139 \def\eq1@numbering@subeq@revert{%
1140 \let\eq1@blocklabel\eq1@parentlabel
1141 \let\eq1@blocklabelname\eq1@parentlabelname
1142 \let\eq1@blocktag\eq1@parenttag
1143 \let\eq1@numbering@subeq@use\eq1@false

```

```

1144 \let\theequation\eql@save@theequation
1145 \let\theHequation\eql@save@theHequation
1146 \eql@subequations@restorecounter
1147 }

```

**TODO:** describe

```

1148 % \TODO note must not use setcounter here (when calc is loaded)
1149 \def\eql@numbering@subeq@close{%
1150   \eql@subequations@close
1151 }

```

## G Display Equations Support

**TODO:** describe

### G.1 Display Breaks

**TODO:** describe

erdisplaylinepenalty

```

1152 \interdisplaylinepenalty\@M

```

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

```

1153 \def\eql@getdsp@pen#1{%
1154   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1155 }

```

**TODO:** allow a displaybreak before equations

```

1156 \protected\def\eql@displaybreak@default{%
1157   \eql@warning{Invalid use of \string\displaybreak}{}%
1158   \eql@teststaroropt@loose\@gobble\eql@gobbleopt{}}
1159 \eql@amsmath@after{\let\eql@displaybreak@default\displaybreak}
1160 \eql@amsmath@let\displaybreak\eql@displaybreak@default

```

```

1161 \newcount\eql@displaybreak@pen@
1162 \newcount\eql@displaybreak@prepen@
1163 \newcount\eql@displaybreak@postpen@

```

**TODO:** describe

```

1164 \protected\def\eql@displaybreak{%
1165   \eql@ampprotecttwo\eql@teststaroropt@tight
1166   \eql@displaybreak@star\eql@displaybreak@opt{4}%
1167 }

```

```

1168 \def\eql@displaybreak@star#1{%
1169   \global\eql@displaybreak@pen@\numexpr#1\relax
1170 }

```

```

1171 \def\eql@displaybreak@opt[#1]{%
1172   \ifnum#1<\z@
1173     \global\eql@displaybreak@pen@\@MM
1174   \else
1175     \global\eql@displaybreak@pen@-\@getpen{#1}\relax

```



```

1176 \fi
1177 }

```

**TODO:** describe

```

1178 \def\eqldisplaybreak@pre#1{%
1179 \ifnum#1<\z@
1180 \eqldisplaybreak@prepen@\@MM
1181 \else
1182 \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1183 \fi
1184 }

```

**TODO:** describe

```

1185 \def\eqldisplaybreak@post#1{%
1186 \ifnum#1<\z@
1187 \eqldisplaybreak@postpen@\@MM
1188 \else
1189 \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1190 \fi
1191 }

```

**TODO:** describe

```

1192 \def\eqldisplaybreak@inter#1{%
1193 \ifnum#1<\z@
1194 \interdisplaylinepenalty\@M
1195 \else
1196 \interdisplaylinepenalty-\eqldgetdsp@pen{#1}\relax
1197 \fi
1198 }

```

## G.2 Vertical Spacing Support

**TODO:** describe

`\eqldstrut` 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.

```

1199 \newbox\eqldstrutbox@
1200 \def\eqldstrut@depth{.3}
1201 \def\eqldstrut{\copy\eqldstrutbox@}
1202 \let\eqldstrut@cell\eqldstrut
1203 \let\eqldstrut@tag\eqldstrut
1204 \def\eqldstrut@make{%
1205 \setbox\eqldstrutbox@\hbox{%
1206 \@tempdimb\dimexpr
1207 \eqldstrut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1208 \@tempdima\dimexpr
1209 \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
1210 \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1211 }
1212 }
1213 \AtBeginDocument{\eqldstrut@make}

```

**TODO:** describe

```

1214 \def\eqldspread@set{%
1215 \eqldspread@\dimexpr\glueexpr\eqldspread@val\relax

```

```

1216      +\normalbaselineskip-\baselineskip\relax
1217 \ifdim\eql@spread@>\z@
1218   \openup\eql@spread@
1219   \ifdefined\spread@equation
1220     \let\spread@equation\@empty
1221   \fi
1222 \fi
1223 }

```

### G.3 Entry and Exit

**TODO:** describe

`\eql@vspaceskip@` (*skip*) **TODO:** add a proper star variant?!

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

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

```

1224 \newskip\eql@vspaceskip@
1225 \newskip\eql@abovespace@
1226 \newskip\eql@belowspace@
1227 \let\eql@vspace@org\vspace
1228 \def\eql@vspace#1\eql@ifstar@loose\eql@vspace@{\eql@vspace@}
1229 \def\eql@vspace@#1{%
1230   \global\advance\eql@vspaceskip@\glueexpr#1\relax}

```

```

1231 \def\eql@display@enter{%
1232   \ifvmode
1233     \eql@prevdepth@\prevdepth
1234     \nointerlineskip
1235     \noindent
1236   \else
1237     \eql@prevdepth@\maxdimen
1238   \fi
1239 }

```

`\eql@display@adjust`

```

1240 \def\eql@display@adjust{%
1241   \ifdefined\eql@display@linewidth
1242     \displaywidth\glueexpr\eql@display@linewidth\relax
1243     \advance\displaywidth-\displayindent
1244   \fi
1245   \ifdefined\eql@display@marginleft
1246     \advance\displaywidth\displayindent
1247     \displayindent\glueexpr\eql@display@marginleft\relax
1248     \advance\displaywidth-\displayindent
1249   \fi
1250   \ifdefined\eql@display@marginright
1251     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1252   \fi
1253   \ifdim\displaywidth<\z@
1254     \displaywidth\z@
1255   \fi
1256 }

```

`\eql@display@init`

```

1257 \def\eql@display@init{%
1258   \let\displaybreak\eql@displaybreak
1259   \let\eql@vspace@org\vspace

```

```

1260 \let\vspace\eql\vspace
1261 \eql@spread@set
1262 \eql@strut@make
1263 }

```

\eql@display@print

```

1264 \def\eql@display@print{%
1265 \eql@display@firstavail@z@
1266 \eql@raisetag@firstlast@z@
1267 \eql@displaybreak@pen@MM
1268 \eql@vspaceskip@z@skip
1269 }

```

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

```

1270 \def\eql@display@halign@init#1{%
1271 \eql@row@z@
1272 \eql@prevgraf@prevgraf
1273 \everycr{\noalign{%
1274 \global\advance\eql@row@one
1275 \prevgraf\numexpr\prevgraf+\one\relax
1276 #1%
1277 }}%
1278 }

```

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

```

1279 \def\eql@display@halign@start{%
1280 \prevgraf\numexpr\eql@prevgraf+\one\relax
1281 \ifdim\eql@prevdepth=\maxdimen\else
1282 \prevdepth\eql@prevdepth@
1283 \fi
1284 \ifdim\prevdepth=-\m@p\else
1285 \ifdefined\eql@display@height
1286 \skip@baselineskip
1287 \advance\skip@-\glueexpr\eql@display@height\relax
1288 \advance\skip@-\prevdepth\relax
1289 \ifdim\skip@<\lineskiplimit
1290 \skip@\lineskip
1291 \fi
1292 \advance\skip@-\eql@spread@\relax
1293 \vskip\skip@
1294 \nointerlineskip
1295 \else
1296 \vskip-\eql@spread@\relax
1297 \fi
1298 \fi
1299 }

```

**TODO:** describe

```

1300 \def\eql@display@vspace{%
1301 \advance\abovedisplayskip\eql@abovespace@
1302 \advance\belowdisplayskip\eql@belowspace@
1303 \advance\belowdisplayskip\eql@vspaceskip@
1304 }

```

**TODO:** describe

```

1305 \def\eql@display@vspace@native{%

```

```

1306 \advance\abovedisplayskip\eql@abovespace@
1307 \advance\belowdisplayskip\eql@belowspace@
1308 \advance\belowdisplayskip\eql@vspaceskip@
1309 \advance\abovedisplayshortskip\eql@abovespace@
1310 \advance\belowdisplayshortskip\eql@belowspace@
1311 \advance\belowdisplayshortskip\eql@vspaceskip@
1312 }

```

**TODO:** describe

```

1313 \def\eql@display@penalty{%
1314 \ifnum\eql@displaybreak@postpen@=\@MM\else
1315 \postdisplaypenalty\eql@displaybreak@postpen@
1316 \fi
1317 \ifnum\eql@displaybreak@pen@=\@MM\else
1318 \postdisplaypenalty\eql@displaybreak@pen@
1319 \fi
1320 \ifnum\eql@displaybreak@prepen@=\@MM\else
1321 \predisplayskip\eql@displaybreak@prepen@
1322 \fi
1323 }

```

**TODO:** describe

```

1324 \def\eql@display@halign@end{%
1325 \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
1326 \ifdefined\eql@display@depth
1327 \prevdepth\glueexpr\eql@display@depth\relax
1328 \fi
1329 }

```

`\eql@display@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?

```

1330 \def\eql@display@close{%
1331 % \TODO temporary fix for development stages
1332 \ifdefined\eql@tagging@on
1333 \ifdefined\dollardollar@begin\else
1334 \belowdisplayskip-\belowdisplayskip
1335 \belowdisplayshortskip-\belowdisplayshortskip
1336 \fi
1337 \fi
1338 \ifdim\eql@display@firstavail@<\z@
1339 \eql@display@firstavail@\z@
1340 \fi
1341 \eql@skip@mode@leave@\z@
1342 \ifdim\eql@prevdepth@=\maxdimen
1343 \ifdim\predisplaysize=-\maxdimen
1344 \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1345 \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1346 \else
1347 \eql@skip@mode@above@\z@
1348 \eql@skip@mode@below@\z@
1349 \advance\eql@display@firstavail@\displayindent
1350 \ifdim\eql@display@firstavail@>\predisplaysize
1351 \ifcase\eql@skip@mode@short\relax
1352 \or
1353 \eql@skip@mode@above@\@ne
1354 \or

```

```

1355         \eqL@skip@mode@above@ \@ne
1356         \ifnum\eqL@totalrows@=\@ne
1357             \eqL@skip@mode@below@ \@ne
1358         \fi
1359     \or
1360         \eqL@skip@mode@above@ \@ne
1361         \eqL@skip@mode@below@ \@ne
1362     \fi
1363 \fi
1364 \fi
1365 \else
1366     \ifdim\eqL@prevdepth@=-\@m\p@
1367         \eqL@skip@mode@above@\eqL@skip@mode@top@above\relax
1368         \eqL@skip@mode@below@\eqL@skip@mode@top@below\relax
1369     \else
1370         \eqL@skip@mode@above@\eqL@skip@mode@par@above\relax
1371         \eqL@skip@mode@below@\eqL@skip@mode@par@below\relax
1372     \fi
1373 \fi
1374 \ifcase\eqL@skip@mode@above@
1375 \or\or\or
1376     \predisplaypenalty\z@
1377 \or
1378     \predisplaypenalty\z@
1379 \fi
1380 \ifcase\eqL@skip@mode@below@
1381 \or\or\or
1382     \eqL@skip@mode@leave@ \@ne
1383 \or
1384     \eqL@skip@mode@leave@ \tw@
1385 \fi
1386 \ifdefined\eqL@skip@force@above
1387     \eqL@skip@mode@above@\eqL@skip@force@above\relax
1388 \fi
1389 \ifdefined\eqL@skip@force@below
1390     \eqL@skip@mode@below@\eqL@skip@force@below\relax
1391 \fi
1392 \ifdefined\eqL@skip@force@leave
1393     \eqL@skip@mode@leave@\eqL@skip@force@leave\relax
1394 \fi
1395 \ifnum\eqL@skip@mode@leave@>\z@
1396     \postdisplaypenalty\z@
1397 \fi
1398 \ifodd\eqL@raisetag@firstlast@
1399     \ifcase\eqL@skip@mode@above@
1400         \abovedisplayskip\glueexpr\eqL@skip@tag@above\relax
1401     \or
1402         \abovedisplayskip\glueexpr\eqL@skip@tag@above\relax
1403     \or
1404         \abovedisplayskip\glueexpr\eqL@skip@tag@above\relax
1405     \or
1406         \abovedisplayskip\glueexpr\eqL@skip@partag@above\relax
1407     \or
1408         \abovedisplayskip\glueexpr\eqL@skip@partag@above\relax
1409     \or
1410         \abovedisplayskip\z@skip
1411     \or
1412         \abovedisplayskip\glueexpr\eqL@skip@medtag@above\relax

```

```

1413 \or
1414 \abovedisplayskip\glueexpr\eq\skip@custom@above\relax
1415 \fi
1416 \else
1417 \ifcase\eq\skip@mode@above@
1418 \abovedisplayskip\glueexpr\eq\skip@long@above\relax
1419 \or
1420 \abovedisplayskip\glueexpr\eq\skip@short@above\relax
1421 \or
1422 \abovedisplayskip\glueexpr\eq\skip@cont@above\relax
1423 \or
1424 \abovedisplayskip\glueexpr\eq\skip@par@above\relax
1425 \or
1426 \abovedisplayskip\glueexpr\eq\skip@top@above\relax
1427 \or
1428 \abovedisplayskip\z@skip
1429 \or
1430 \abovedisplayskip\glueexpr\eq\skip@med@above\relax
1431 \or
1432 \abovedisplayskip\glueexpr\eq\skip@custom@above\relax
1433 \fi
1434 \fi
1435 \ifnum\eq\raisetag@firstlast@>\@ne
1436 \ifcase\eq\skip@mode@below@
1437 \belowdisplayskip\glueexpr\eq\skip@tag@below\relax
1438 \or
1439 \belowdisplayskip\glueexpr\eq\skip@tag@below\relax
1440 \or
1441 \belowdisplayskip\glueexpr\eq\skip@tag@below\relax
1442 \or
1443 \belowdisplayskip\glueexpr\eq\skip@partag@below\relax
1444 \or
1445 \belowdisplayskip\glueexpr\eq\skip@partag@below\relax
1446 \or
1447 \belowdisplayskip\z@skip
1448 \or
1449 \belowdisplayskip\glueexpr\eq\skip@medtag@below\relax
1450 \or
1451 \belowdisplayskip\glueexpr\eq\skip@custom@below\relax
1452 \fi
1453 \else
1454 \ifcase\eq\skip@mode@below@
1455 \belowdisplayskip\glueexpr\eq\skip@long@below\relax
1456 \or
1457 \belowdisplayskip\glueexpr\eq\skip@short@below\relax
1458 \or
1459 \belowdisplayskip\glueexpr\eq\skip@cont@below\relax
1460 \or
1461 \belowdisplayskip\glueexpr\eq\skip@par@below\relax
1462 \or
1463 \belowdisplayskip\glueexpr\eq\skip@top@below\relax
1464 \or
1465 \belowdisplayskip\z@skip
1466 \or
1467 \belowdisplayskip\glueexpr\eq\skip@med@below\relax
1468 \or
1469 \belowdisplayskip\glueexpr\eq\skip@custom@below\relax
1470 \fi

```

```

1471 \fi
1472 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
1473 \eq@display@penalty
1474 \eq@display@vspace
1475 % \TODO temporary fix for development stages
1476 \ifdefined\eq@tagging@on
1477   \ifdefined\dollar@begin@else
1478     \belowdisplayskip-\belowdisplayskip
1479   \fi
1480 \fi
1481 }

```

**TODO:** describe

```

1482 \def\eq@display@leave{%
1483   \prevgraf\eq@prevgraf@
1484   \ifcase\eq@skip@mode@leave@
1485   \or
1486     \endgraf
1487   \or
1488     \endgraf
1489     \prevdepth-\@m\p@
1490 \fi
1491 }

```

**TODO:** describe

```

1492 \def\eq@display@nest{%
1493   \let\label\eq@label@org
1494   \let\tag\eq@tag@default
1495   \let\raisetag\eq@raisetag@default
1496   \let\displaybreak\eq@displaybreak@default
1497   \let\intertext\eq@intertext@default
1498   \let\vspace\eq@vspace@org
1499 }

```

**TODO:** describe

```

1500 \expandafter\def\expandafter\arrayparboxrestore\expandafter{%
1501   \@arrayparboxrestore
1502   \eq@display@nest
1503   \ifdefined\eq@ampproof@active
1504     \eq@ampprevert
1505   \fi
1506   \@displayfalse
1507 }

```

## G.4 Stack

**TODO:** describe

```

1508 \def\eq@stack@enable{%
1509   \let\eq@stack@save@single\eq@stack@save@single@
1510   \let\eq@stack@save@multi\eq@stack@save@multi@
1511   \let\eq@stack@save@boxed\eq@stack@save@boxed@
1512 }

```

**TODO:** describe

```

1513 \let\eq@stack@save@single\eq@stack@enable

```

```

1514 \let\eqL@stack@save@multi\eqL@stack@enable
1515 \let\eqL@stack@save@boxed\eqL@stack@enable
1516 \let\eqL@stack@restore\@empty

```

**TODO:** describe

```

1517 \def\eqL@stack@save@reg#1{\global#1\the#1\relax}
1518 \def\eqL@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

**TODO:** describe

```

1519 \def\eqL@stack@save@single@{%
1520   \let\eqL@stack@nextlabel\eqL@nextlabel
1521   \let\eqL@stack@nextlabelname\eqL@nextlabelname
1522   \let\eqL@stack@nexttag\eqL@nexttag
1523   \edef\eqL@stack@restore{%
1524     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1525     \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1526     \eqL@stack@save@let\eqL@stack@nextlabelname\eqL@nextlabelname
1527     \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1528     \eqL@stack@save@let\eqL@stack@frame@cmd\eqL@frame@cmd
1529     \eqL@stack@save@reg\eqL@displaybreak@pen@
1530     \eqL@stack@save@reg\eqL@vspaceskip@
1531     \eqL@stack@save@reg\eqL@shape@pos@
1532     \eqL@stack@save@reg\eqL@shape@amount@
1533     \eqL@stack@save@reg\eqL@display@firstavail@
1534     \eqL@stack@save@reg\eqL@raisetag@amount@
1535     \eqL@stack@save@reg\eqL@raisetag@firstlast@
1536     \eqL@stack@save@reg\eqL@row@
1537   }%
1538 }

```

**TODO:** describe

```

1539 \def\eqL@stack@save@multi@{%
1540   \let\eqL@stack@nextlabel\eqL@nextlabel
1541   \let\eqL@stack@nextlabelname\eqL@nextlabelname
1542   \let\eqL@stack@nexttag\eqL@nexttag
1543   \let\eqL@stack@widthdata@tab\eqL@widthdata@tab
1544   \let\eqL@stack@label@thepage\eqL@label@thepage
1545   \let\eqL@stack@label@currentHref\eqL@label@currentHref
1546   \edef\eqL@stack@restore{%
1547     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1548     \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1549     \eqL@stack@save@let\eqL@stack@nextlabelname\eqL@nextlabelname
1550     \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1551     \eqL@stack@save@let\eqL@stack@widthdata@tab\eqL@widthdata@tab
1552     \eqL@stack@save@let\eqL@stack@label@thepage\eqL@label@thepage
1553     \eqL@stack@save@let\eqL@stack@label@currentHref\eqL@label@currentHref
1554     \eqL@stack@save@let\eqL@stack@frame@cmd\eqL@frame@cmd
1555     \eqL@stack@save@reg\eqL@displaybreak@pen@
1556     \eqL@stack@save@reg\eqL@vspaceskip@
1557     \eqL@stack@save@reg\eqL@shape@pos@
1558     \eqL@stack@save@reg\eqL@shape@amount@
1559     \eqL@stack@save@reg\eqL@display@firstavail@
1560     \eqL@stack@save@reg\eqL@raisetag@amount@
1561     \eqL@stack@save@reg\eqL@raisetag@firstlast@
1562     \eqL@stack@save@reg\eqL@column@
1563     \eqL@stack@save@reg\eqL@totalcolumns@
1564     \eqL@stack@save@reg\eqL@line@avail@
1565     \eqL@stack@save@reg\eqL@line@pos@

```



```

1566 \eqL@stack@save@reg\eqL@line@width@
1567 \eqL@stack@save@reg\eqL@line@depth@
1568 \eqL@stack@save@reg\eqL@line@height@
1569 \eqL@stack@save@reg\eqL@tagwidth@max@
1570 \eqL@stack@save@reg\eqL@numbering@target@
1571 \eqL@stack@save@reg\eqL@row@
1572 \eqL@stack@save@reg\eqL@tagrows@
1573 }%
1574 }
1575 \def\eqL@stack@save@boxed@{%
1576 \edef\eqL@stack@restore{%
1577 \eqL@stack@save@reg\eqL@row@
1578 \eqL@stack@save@reg\eqL@totalrows@
1579 \eqL@stack@save@reg\eqL@shape@pos@
1580 \eqL@stack@save@reg\eqL@shape@amount@
1581 \eqL@stack@save@let\eqL@stack@frame@cmd\eqL@frame@cmd
1582 }%
1583 }

```

## H Multi-Line Support

**TODO:** describe

### H.1 Measure Support

**TODO:** describe

```

1584 \def\eqL@measure@init#1#2{%
1585 \eqL@widthdata@reset
1586 \eqL@numbering@measure@init
1587 \eqL@row@z@
1588 \tabskip\z@skip
1589 \everycr{\noalign{%
1590 \global\advance\eqL@row@\@ne
1591 #1%
1592 }}%
1593 \eqL@measure@savestate
1594 \measuring@true
1595 \eqL@display@halign@letcr{#2}%
1596 }

1597 \def\eqL@measure@close{%
1598 \advance\eqL@row@-\tw@
1599 \eqL@totalrows@\eqL@row@
1600 \eqL@numbering@measure@eval
1601 \ifnum\eqL@numbering@target@>\z@
1602 \eqL@tagbox@make\eqL@compose@measure
1603 \eqL@widthdata@savetagsingle
1604 \fi
1605 \eqL@measure@restorestate
1606 }

```

measure@restorestate

eqL@measure@savestate

```
1607 \let\eqL@measure@restorestate\@empty
```

```
1608 \def\eqL@measure@savestate{%

```

```

1609 \begingroup
1610   \def\@elt##1{%
1611     \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
1612   \global\edef\@gtempa{%
1613     \cl@ckpt
1614     \let\noexpand\eq@measure@restorestate\noexpand\@empty
1615     \ifmeasuring@\noexpand\measuring@true\else\noexpand\measuring@false\fi
1616   }%
1617 \endgroup
1618 \let\eq@measure@restorestate\@gtempa
1619 }

```

## H.2 Line Breaks

**TODO:** describe

`\eq@display@cr`

```

1620 \protected\def\eq@display@cr{%
1621   \eq@ampprotecttwo\eq@teststaropt@tight
1622   {\global\eq@displaybreak@pen@\@M\eq@display@cr@opt}
1623   \eq@display@cr@opt\z@
1624 }

```

`\eq@display@cr@opt`

```

1625 \def\eq@display@cr@opt[#1]{%
1626   \eq@display@endline
1627   \cr
1628   \noalign{%
1629     \ifnum\eq@displaybreak@pen@=\@MM
1630       \penalty\interdisplaylinepenalty
1631       \global\eq@displaybreak@pen@\@MM
1632     \else
1633       \penalty\eq@displaybreak@pen@
1634     \fi
1635     \advance\eq@vspaceskip@\glueexpr#1\relax%
1636     \vskip\eq@vspaceskip@
1637     \global\eq@vspaceskip@\z@skip
1638   }%
1639 }

```

`display@halign@letcr`

```

1640 \def\eq@display@halign@letcr#1{%
1641   \let\\\eq@display@cr
1642   \let\eq@display@endline#1%
1643 }

```

## H.3 Intertext

**TODO:** describe

**TODO:** revert in everymath?

```

1644 \def\eq@intertext@default{\eq@error{Invalid use of \string\intertext}}
1645 \eq@amsmath@let\intertext\eq@intertext@default

```

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

```

1646 \def\eql@intertext@process{%
1647   \eql@display@endline
1648   \cr
1649   \ifmeasuring@
1650     \expandafter\@gobble
1651   \else
1652     \expandafter\eql@intertext@print
1653   \fi
1654 }

```

**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

```

1655 \def\eql@intertext@print#1{%
1656   \noalign{%
1657     \eql@display@halign@end
1658     \let\eql@skip@force@below\z@
1659     \let\eql@skip@force@above\z@
1660     \eql@setkeys{intertext}\eql@intertext@opt
1661     \openup-\eql@spread@
1662     \penalty\postdisplaypenalty
1663     \ifcase\eql@skip@force@below\relax
1664       \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
1665     \or
1666       \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
1667     \or
1668       \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
1669     \or
1670       \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
1671     \or
1672       \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
1673     \or
1674       \advance\eql@vspaceskip@\z@skip
1675     \or
1676       \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
1677     \or
1678       \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
1679     \fi
1680     \vskip\eql@vspaceskip@
1681     \global\eql@vspaceskip@\z@skip
1682     \vbox{%
1683       \@parboxrestore
1684       \ifdim
1685         \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
1686       \else
1687         \parshape\@ne
1688         \@totalleftmargin\linewidth
1689       \fi
1690       \noindent
1691       \prevgraf\eql@prevgraf@
1692       \ignorespaces
1693       #1%
1694       \par
1695       \global\eql@prevgraf@\prevgraf
1696     }%
1697     \penalty\predisplaypenalty
1698     \ifcase\eql@skip@force@above\relax

```

```

1699     \vskip\glueexpr\eq\@skip@long@above\relax
1700   \or
1701     \vskip\glueexpr\eq\@skip@short@above\relax
1702   \or
1703     \vskip\glueexpr\eq\@skip@cont@above\relax
1704   \or
1705     \vskip\glueexpr\eq\@skip@par@above\relax
1706   \or
1707     \vskip\glueexpr\eq\@skip@top@above\relax
1708   \or
1709     \vskip\z@skip
1710   \or
1711     \vskip\glueexpr\eq\@skip@med@above\relax
1712   \or
1713     \vskip\glueexpr\eq\@skip@custom@above\relax
1714   \fi
1715 %   \eq\@prevdepth@maxdimen
1716   \eq\@prevdepth@z@
1717   \eq\@display@halign@start
1718 }
1719 }

```

**TODO:** describe

```

1720 \newenvironment{eq\@intertext}{%
1721   \eq\@testopt@tight\eq\@intertext@{}}%
1722 }{%
1723   \aftergroup\eq\@intertext@after
1724   \ignorespacesafterend
1725 }

```

**TODO:** describe

```

1726 \def\eq\@intertext@env{intertext}
1727 \def\eq\@intertext@[#1]{%
1728   \global\def\eq\@intertext@opt{#1}%
1729   \ifx\@currenvir\eq\@intertext@env
1730     \expandafter\eq\@scan@env\expandafter\eq\@intertext@inject
1731   \else
1732     \expandafter\eq\@intertext@process
1733   \fi
1734 }

```

**TODO:** describe

```

1735 \def\eq\@intertext@inject{%
1736   \global\edef\eq\@intertext@after{%
1737     \noexpand\eq\@intertext@process{%
1738       \ifx\eq\@scan@body\eq\@scan@body@dump
1739         \eq\@scan@body@dump
1740       \else
1741         \noexpand\scantokens{\eq\@scan@body@dump}%
1742       \fi
1743     }%
1744   }%
1745 }

```

# 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:

```
1746 \newcount\eq@shape@pos@
1747 \newdimen\eq@shape@amount@
```

`\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*)

```
1748 \newdimen\eq@marginleft@
1749 \newdimen\eq@marginright@
1750 \newdimen\eq@marginleft@min@
1751 \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:

```
1752 \def\eq@shape@tab@default{default}
1753 \def\eq@shape@tab@left{left}
1754 \def\eq@shape@tab@center{center}
1755 \def\eq@shape@tab@right{right}
1756 \def\eq@shape@tab@first{first}
1757 \def\eq@shape@tab@hanging{hanging}
1758 \def\eq@shape@tab@steps{steps}
```

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

```
1759 \let\eq@shape@tab@def\eq@shape@tab@default
1760 \let\eq@shape@tab@\eq@shape@tab@default
1761 \let\eq@shape@tab@l\eq@shape@tab@left
1762 \let\eq@shape@tab@c\eq@shape@tab@center
1763 \let\eq@shape@tab@r\eq@shape@tab@right
1764 \let\eq@shape@tab@rc\eq@shape@tab@first
1765 \let\eq@shape@tab@indent\eq@shape@tab@first
1766 \let\eq@shape@tab@hang\eq@shape@tab@hanging
1767 \let\eq@shape@tab@lc\eq@shape@tab@hanging
1768 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
1769 \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:

```
1770 \let\eq@shape@mode\eq@shape@tab@default
```

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

```
1771 \def\eq@shape@set#1{%
1772   \ifcsname eq@shape@tab@#1\endcsname
```

```

1773 \expandafter\let\expandafter\eq@shape@mode
1774 \csname eq@shape@tab@#1\endcsname
1775 \else
1776 \eq@error{shape ‘#1’ unknown: setting to default}%
1777 \let\eq@shape@mode\eq@shape@tab@default
1778 \fi
1779 }

```

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

```

1780 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
1781 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne\eq@shape@amount@z@}
1782 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
1783 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
1784 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
1785 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne\eq@shape@amount@z@}
1786 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
1787 \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):

```

1788 \def\eq@shape@layoutcenter@first{%
1789 \eq@shape@pos@z@
1790 \eq@shape@amount@z@
1791 \ifnum\eq@totalrows@>\@ne
1792 \ifnum\eq@row@=\@ne
1793 \eq@shape@amount@\eq@indent@
1794 \fi
1795 \fi
1796 }
1797 \def\eq@shape@layoutleft@first{%
1798 \eq@shape@pos@z@
1799 \eq@shape@amount@z@
1800 \ifnum\eq@totalrows@>\@ne
1801 \ifnum\eq@row@=\@ne
1802 \eq@shape@amount@\eq@indent@
1803 \fi
1804 \fi
1805 }

```

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:

```

1806 \def\eq@shape@layoutcenter@hanging{%
1807 \eq@shape@pos@z@
1808 \eq@shape@amount@\eq@indent@
1809 \ifnum\eq@totalrows@>\@ne
1810 \ifnum\eq@row@=\@ne
1811 \eq@shape@amount@z@
1812 \fi
1813 \fi
1814 }
1815 \def\eq@shape@layoutleft@hanging{%
1816 \eq@shape@pos@z@
1817 \eq@shape@amount@z@
1818 \ifnum\eq@totalrows@>\@ne

```

```

1819 \ifnum\eq@row@=\@ne
1820 \eq@shape@amount@-\eq@indent@
1821 \fi
1822 \fi
1823 }

```

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:

```

1824 \def\eq@shape@layoutcenter@steps{%
1825 \eq@shape@amount@z@
1826 \eq@shape@pos@\@ne
1827 \ifnum\eq@totalrows@>\@ne
1828 \ifnum\eq@row@=\@ne
1829 \eq@shape@pos@z@
1830 \fi
1831 \ifnum\eq@row@=\eq@totalrows@
1832 \eq@shape@pos@tw@
1833 \fi
1834 \fi
1835 }
1836 \def\eq@shape@layoutleft@steps{%
1837 \eq@shape@pos@z@
1838 \eq@shape@amount@z@
1839 \ifnum\eq@totalrows@>\@ne
1840 \ifnum\eq@row@=\@ne
1841 \eq@shape@amount@-\eq@indent@
1842 \fi
1843 \ifnum\eq@row@=\eq@totalrows@
1844 \eq@shape@amount@\eq@indent@
1845 \fi
1846 \fi
1847 }

```

`\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`:

```

1848 \let\eq@shape@eval\@undefined
1849 \def\eq@shape@select{%
1850 \expandafter\let\expandafter\eq@shape@eval
1851 \csname eq@shape%
1852 @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
1853 @\eq@shape@mode\endcsname
1854 }

```

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

`\eq@shape@alignright`

`\eq@shape@aligncenter`

```

1855 \protected\def\eq@shape@alignleft{%
1856 \global\eq@shape@pos@z@
1857 \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
1858 \protected\def\eq@shape@aligncenter{%
1859 \global\eq@shape@pos@\@ne
1860 \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
1861 \protected\def\eq@shape@alignright{%
1862 \global\eq@shape@pos@tw@
1863 \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
1864 \def\eq@shape@align@testpar#1{%

```

```

1865 \eq@ifstar@tight{#1[\eq@indent0]}%
1866 {\eq@ifnextgobble@tight{!}{#1[-\eq@indent0]}%
1867 {\eq@testopt@tight{#1}\z@}}
1868 \def\eq@shape@alignamount@opt[#1]{%
1869 \global\eq@shape@amount@\glueexpr#1\relax}

```

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

```

1870 \protected\def\eq@shape@alignamount{%
1871 \eq@ampprotecttwo\eq@ifstar@tight
1872 \eq@shape@alignamount@set\eq@shape@alignamount@add}
1873 \def\eq@shape@alignamount@add#1{%
1874 \ifnum\eq@shape@pos@=\m@ne
1875 \global\eq@shape@pos@-\thr@@
1876 \eq@shape@amount@\z@
1877 \fi
1878 \global\advance\eq@shape@amount@\glueexpr#1\relax}
1879 \def\eq@shape@alignamount@set#1{%
1880 \ifnum\eq@shape@pos@<\z@
1881 \global\eq@shape@pos@-\tw@
1882 \fi
1883 \global\eq@shape@amount@\glueexpr#1\relax}
1884 \def\eq@shape@align@enable{%
1885 \let\shoveleft\eq@shape@alignleft
1886 \let\shovecenter\eq@shape@aligncenter
1887 \let\shoveright\eq@shape@alignright
1888 \let\shoveby\eq@shape@alignamount
1889 }

```

**TODO:** describe

```

1890 \protected\def\eq@shape@align@default{%
1891 \eq@warning{\string\shove... not allowed here}%
1892 \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
1893 \protected\def\eq@shape@alignamount@default{%
1894 \eq@warning{\string\shove... not allowed here}%
1895 \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
1896 \def\eq@shape@align@disable{%
1897 \let\shoveleft\eq@shape@align@default
1898 \let\shovecenter\eq@shape@align@default
1899 \let\shoveright\eq@shape@align@default
1900 \let\shoveby\eq@shape@alignamount@default
1901 }

```

## I.3 Width Data

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

```

1902 \newdimen\eq@tagwidth@single@

```

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

```

1903 \let\eq@widthdata@tab\@empty

```

`\eq@widthdata@reset`

```

1904 \def\eq@widthdata@reset{%
1905 \let\eq@widthdata@tab\@empty
1906 \eq@tagwidth@max@\z@

```



```

1907 \eql@tagrows@\z@
1908 }

\eql@widthdata@add

1909 \def\eql@widthdata@add#1{%
1910 \expandafter\global\expandafter\def\expandafter\eql@widthdata@tab
1911 \expandafter{\eql@widthdata@tab#1}%
1912 }

l@widthdata@startrow

1913 \def\eql@widthdata@startrow{%
1914 \expandafter\eql@widthdata@add\expandafter{%
1915 \expandafter\eql@row@\the\eql@row@\relax}%
1916 }

l@widthdata@savecell

1917 \def\eql@widthdata@savecell{%
1918 \edef\eql@tmp{%
1919 \eql@shape@pos@\the\eql@shape@pos@\relax
1920 \eql@cellwidth@\the\eql@cellwidth@\relax
1921 \eql@shape@amount@\the\eql@shape@amount@\relax
1922 \noexpand\eql@widthdata@cellcall
1923 }%
1924 \expandafter\eql@widthdata@add\expandafter{\eql@tmp}%
1925 }

ql@widthdata@savesep

1926 \def\eql@widthdata@savesep{%
1927 \eql@widthdata@add\eql@widthdata@sepcall
1928 }

ql@widthdata@savetag

1929 \def\eql@widthdata@savetag{%
1930 \expandafter\eql@widthdata@add\expandafter{%
1931 \expandafter\eql@tagwidth@\the\eql@tagwidth@\relax;}%
1932 \ifdim\eql@tagwidth@>\eql@tagwidth@max@
1933 \global\eql@tagwidth@max@\eql@tagwidth@
1934 \fi
1935 \ifdim\eql@tagwidth@>\z@
1936 \global\advance\eql@tagrows@\@ne
1937 \fi
1938 }

thdata@savetagsingle

1939 \def\eql@widthdata@savetagsingle{%
1940 \eql@tagwidth@single@\eql@tagwidth@
1941 \eql@tagwidth@max@\eql@tagwidth@
1942 \eql@tagrows@\@ne
1943 }

\eql@widthdata@for

1944 \def\eql@widthdata@for#1{%

```

```

1945 \def\eql@widthdata@forall{#1}%
1946 \expandafter\eql@widthdata@forstep\eql@widthdata@tab
1947 \eql@row@0\relax\eql@tagwidth@z\relax;%
1948 }

```

eql@widthdata@forstep

```

1949 \def\eql@widthdata@forstep\eql@row@#1\relax#2\eql@tagwidth@#3\relax;{%
1950 \eql@row@#1\relax
1951 \ifnum\eql@row@=z@\else
1952 \eql@tagwidth@#3\relax
1953 \def\eql@widthdata@cells{#2}%
1954 \eql@widthdata@forall
1955 \expandafter\eql@widthdata@forstep
1956 \fi
1957 }

```

\eql@widthdata@get

```

1958 \def\eql@widthdata@get#1{%
1959 \eql@row@#1\relax
1960 \expandafter\eql@widthdata@getdef\expandafter{\the\eql@row@}%
1961 \expandafter\eql@widthdata@getparse\eql@widthdata@tab@nil%
1962 }

```

eql@widthdata@getdef

```

1963 \def\eql@widthdata@getdef#1{%
1964 \def\eql@widthdata@getparse
1965 ##1\eql@row@#1\relax##2\eql@tagwidth@##3\relax;##4@nil{%
1966 \eql@tagwidth@##3\relax
1967 \def\eql@widthdata@cells{##2}%
1968 }%
1969 }

```

\eql@colwidth@tab

```

1970 \let\eql@colwidth@tab\empty

```

\eql@colwidth@get

```

1971 \def\eql@colwidth@get#1{%
1972 \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
1973 }

```

\eql@colwidth@save

```

1974 \def\eql@colwidth@save#1{%
1975 \edef\eql@tmp{\noexpand\or\the#1}%
1976 \expandafter\expandafter\expandafter\def
1977 \expandafter\expandafter\expandafter\eql@colwidth@tab
1978 \expandafter\expandafter\expandafter{%
1979 \expandafter\eql@tmp\eql@colwidth@tab}%
1980 }

```

\eql@widthdata@calc Compute the space that is available at the beginning and at the end of the row stored in \eql@widthdata@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@`:

```

1981 \def\eql@widthdata@calc{%
1982   \eql@column@\z@
1983   \eql@line@pos@\z@
1984   \eql@line@possep@\z@
1985   \eql@line@avail@\eql@totalwidth@
1986   \eql@line@availsep@\eql@columns@inter@
1987   \eql@line@width@\z@
1988   \eql@line@widthsep@\z@
1989   \let\eql@widthdata@cellcall\eql@widthdata@calc@call
1990   \let\eql@widthdata@sepcall\eql@widthdata@calc@callsep
1991   \eql@widthdata@cells
1992 }
```

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

```

1993 \def\eql@widthdata@calc@callsep{%
1994   \advance\eql@line@possep@\@ne
1995 }%
```

`@widthdata@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?!)

```

1996 \def\eql@widthdata@calc@call{%
1997   \advance\eql@column@\@ne
1998   \ifnum\eql@totalcolumns@=\@ne
1999     \dimen@\eql@totalwidth@
2000   \else
2001     \dimen@\eql@colwidth@get\eql@column@\relax
2002   \fi
2003   \ifdim\eql@cellwidth@>\z@
2004     \ifdim\eql@line@width@=\z@
2005       \eql@line@avail@\eql@line@pos@
2006       \eql@line@availsep@\eql@line@possep@
2007       \ifcase\eql@shape@pos@
2008         \or
2009           \advance\eql@line@avail@\dimexpr
2010             (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2011         \or
2012           \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
2013         \fi
2014       \advance\eql@line@avail@\eql@shape@amount@
2015     \fi
2016     \eql@line@width@\eql@line@pos@
2017     \eql@line@widthsep@\eql@line@possep@
2018     \ifcase\eql@shape@pos@
2019       \advance\eql@line@width@\eql@cellwidth@
2020     \or
2021       \advance\eql@line@width@\dimexpr
2022         (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2023     \or
2024       \advance\eql@line@width@\dimen@
2025     \fi
2026     \advance\eql@line@width@\eql@shape@amount@
2027   \fi
2028   \advance\eql@line@pos@\dimen@
```

2029 }

## I.4 Best Line Selection

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

```
2030 \let\eql@numbering@best@auto\eql@false
```

@g@best@row@ (*counter*)

@g@best@space@ (*dimen*)

@numbering@best@use (*bool*)

```
2031 \newcount\eql@numbering@best@row@
```

```
2032 \newdimen\eql@numbering@best@space@
```

```
2033 \let\eql@numbering@best@use\eql@false
```

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

```
2034 \def\eql@numbering@best@find{%
2035   \eql@numbering@best@row@ \z@
2036   \eql@numbering@best@space@ \z@
2037   \eql@widthdata@for{%
2038     \eql@widthdata@calc
2039     \ifdefined\eql@tagsleft
2040       \dimen@ \eql@line@avail@
2041     \else
2042       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
2043     \fi
2044     \ifdim \dimen@ > \eql@numbering@best@space@
2045       \eql@numbering@best@row@ \eql@row@
2046       \eql@numbering@best@space@ \dimen@
2047     \fi
2048   }%
2049   \ifnum \eql@numbering@best@row@ > \z@
2050     \eql@numbering@target@ \eql@numbering@best@row@
2051   \fi
2052 }
```

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

```
2053 \def\eql@numbering@best@test{%
2054   \eql@widthdata@get \eql@numbering@target@
2055   \eql@widthdata@calc
2056   \ifdefined\eql@tagsleft
2057     \dimen@ \dimexpr \eql@line@avail@
2058       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
2059   \else
2060     \dimen@ \dimexpr \displaywidth - \eql@line@width@
2061       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
2062   \fi
2063   \ifdim \dimen@ < \eql@tagwidth@single@
2064     \let \eql@numbering@best@use \eql@true
2065   \fi
2066 }
```

@numbering@best@eval **TODO:** describe

```
2067 \def\eql@numbering@best@eval{%
2068   \ifdefined\eql@numbering@best@auto
2069     \ifdefined\eql@numbering@best@use \else
```

```

2070     \ifnum\eqL@numbering@target@>\z@
2071     \eqL@numbering@best@test
2072     \fi
2073     \fi
2074     \fi
2075     \ifdefined\eqL@numbering@best@use
2076     \eqL@numbering@best@find
2077     \fi
2078 }

```

## 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

```

2079 \def\eqL@adjust@calc@tagmargin{%
2080   \ifdefined\eqL@tagmargin@val
2081     \eqL@tagmargin@\glueexpr\eqL@tagmargin@val\relax
2082   \else
2083     \eqL@tagmargin@\eqL@tagwidth@max@
2084     \ifdim\eqL@tagmargin@>\z@
2085       \advance\eqL@tagmargin@-\eqL@tagsepmin@
2086     \fi
2087   \fi

2088   \dimen@\eqL@tagrows@p@
2089   \ifnum\eqL@totalrows@=\@ne
2090     \ifnum\eqL@tagrows@=\@ne
2091       \advance\dimen@1sp\relax
2092     \fi
2093   \fi
2094   \ifdim\dimen@>\eqL@totalrows@\eqL@tagmargin@ratio@\else
2095     \eqL@tagmargin@\z@
2096   \fi

2097   \@tempdima\dimexpr\displaywidth
2098     -\eqL@totalwidth@-\eqL@columns@inter@\eqL@colsepmin@\relax
2099   \@tempdimb\dimexpr\@tempdima-\tw@\eqL@tagmargin@\relax
2100   \ifdim\@tempdimb>\z@
2101     \ifdim\eqL@tagmargin@threshold\@tempdima<\@tempdimb
2102       \eqL@tagmargin@\z@
2103     \fi
2104   \fi
2105 }

```

## I.6 Single Column

ql@adjust@calc@lines

```

2106 \def\eqL@adjust@calc@lines{%
2107   \eqL@totalcolumns@\@ne
2108   \eqL@columns@inter@\z@
2109   \eqL@colsep@\z@
2110   \ifdefined\eqL@layoutleft
2111     \eqL@marginleft@\glueexpr\eqL@layoutleftmargin\relax

```

```

2112 \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2113 \ifdim\eql@marginleft@<\eql@marginleft@min@
2114 \eql@marginleft@\eql@marginleft@min@
2115 \fi
2116 \dimen@\glueexpr\eql@layoutleftmarginmax\relax
2117 \ifdim\eql@marginleft@>\dimen@
2118 \eql@marginleft@\dimen@
2119 \fi
2120 \eql@marginright@z@
2121 \eql@centeroffset@z@
2122 \else
2123 \eql@adjust@calc@tagmargin
2124 \ifdefined\eql@paddingleft@val
2125 \eql@marginleft@\dimexpr
2126 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2127 -\glueexpr\eql@paddingleft@val\relax\relax
2128 \ifdim\eql@marginleft@<z@
2129 \eql@marginleft@z@
2130 \fi
2131 \else
2132 \eql@marginleft@z@
2133 \fi
2134 \ifdefined\eql@paddingright@val
2135 \eql@marginright@\dimexpr
2136 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2137 -\glueexpr\eql@paddingright@val\relax\relax
2138 \ifdim\eql@marginright@<z@
2139 \eql@marginright@z@
2140 \fi
2141 \else
2142 \eql@marginright@z@
2143 \fi
2144 \ifdim\eql@tagmargin@>z@
2145 \ifdefined\eql@tagsleft
2146 \ifdim\eql@marginleft@<\eql@tagsepmin@
2147 \eql@marginleft@\eql@tagsepmin@
2148 \fi
2149 \advance\eql@marginleft@\eql@tagmargin@
2150 \advance\eql@centeroffset@\eql@tagmargin@
2151 \else
2152 \ifdim\eql@marginright@<\eql@tagsepmin@
2153 \eql@marginright@\eql@tagsepmin@
2154 \fi
2155 \advance\eql@marginright@\eql@tagmargin@
2156 \advance\eql@centeroffset@-\eql@tagmargin@
2157 \fi
2158 \fi
2159 \eql@marginleft@min@z@
2160 \eql@centeroffset@\dimexpr\eql@marginright@-\eql@marginleft@
2161 \ifdefined\eql@tagsleft+\else-\fi\eql@tagmargin@\relax
2162 \fi

2163 \eql@totalwidth@\dimexpr\displaywidth
2164 -\eql@marginleft@-\eql@marginright@\relax
2165 }

```

## 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:

```
2166 \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:

```
2167 \ifnum\eql@totalcolumns@<\thr@@
2168   \eql@totalcolumns@=\tw@
2169   \let\eql@columns@fulllength\eql@false
2170 \fi
```

Determine the number of intercolumn spaces `\eql@columns@inter@`:

```
2171 \eql@columns@inter@=\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2172 \eql@colsepmin@=\glueexpr\eql@colsepmin@val\relax
```

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

```
2173 \ifdefined\eql@layoutleft
2174   \eql@marginleft@=\glueexpr\eql@layoutleftmargin\relax
2175   \eql@marginleft@min@=\glueexpr\eql@layoutleftmarginmin\relax
2176   \ifdim\eql@marginleft@<\eql@marginleft@min@
2177     \eql@marginleft@=\eql@marginleft@min@
2178   \fi
2179 \else
```

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

```
2180   \eql@adjust@calc@tagmargin
2181   \ifdefined\eql@columns@fulllength
2182     \ifdim\eql@tagmargin@>\z@
2183       \advance\eql@tagmargin@=\eql@tagsepmin@
2184     \fi
2185   \fi
2186   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2187     -\eql@columns@inter@\eql@colsepmin@\relax
2188     \eql@tagmargin@=\z@
2189   \fi
2190   \eql@marginleft@min@=\z@
2191 \fi
```

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

```
2192 \ifnum\eql@columns@inter@>\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:

```

2193 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2194 \ifdefined\eql@layoutleft
2195 \advance\eql@colsep@-\eql@marginleft@
2196 \else
2197 \advance\eql@colsep@-\eql@tagmargin@
2198 \fi
2199 \count@\eql@columns@inter@
2200 \ifdefined\eql@columns@fulllength\else
2201 \ifdefined\eql@layoutleft
2202 \advance\count@\@ne
2203 \else
2204 \advance\count@\tw@
2205 \fi
2206 \fi
2207 \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:

```

2208 \ifdim\eql@colsep@<\eql@colsepmin@
2209 \eql@colsep@\eql@colsepmin@
2210 \else
2211 \ifdefined\eql@columns@fulllength\else
2212 \dimen@\glueexpr\eql@colsepmax@val\relax
2213 \ifdim\eql@colsep@>\dimen@
2214 \eql@colsep@\dimen@
2215 \fi
2216 \fi
2217 \fi
2218 \else

```

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

```

2219 \eql@colsep@\eql@colsepmin@
2220 \fi

```

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

```

2221 \ifdefined\eql@layoutleft

```

Set the default value:

```

2222 \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:

```

2223 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
2224 -\eql@columns@inter@\eql@colsep@\relax
2225 \ifdim\dimen@<\eql@marginleft@
2226 \ifdim\dimen@<\eql@marginleft@min@
2227 \eql@marginleft@\eql@marginleft@min@
2228 \else
2229 \eql@marginleft@\dimen@
2230 \fi
2231 \fi
2232 \fi
2233 \else

```

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



```

2234 \ifdefined\eql@columns@fulllength
2235 \eql@marginleft@<z@

```

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

```

2236 \else
2237 \eql@marginleft@<\dimexpr(\displaywidth-\eql@totalwidth@
2238 -\eql@columns@inter@<\eql@colsep@-\eql@tagmargin@)/\tw@<\relax
2239 \ifdim\eql@marginleft@<<z@
2240 \eql@marginleft@<z@
2241 \fi
2242 \fi

```

Add tag margin in case of left tags:

```

2243 \ifdefined\eql@tagsleft
2244 \advance\eql@marginleft@<\eql@tagmargin@
2245 \fi
2246 \fi

```

Find the best row for tag placement:

```

2247 \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:

```

2248 \ifdefined\eql@tagsleft
2249 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
2250 \else
2251 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2252 \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:

```

2253 \ifnum\eql@numbering@target@<<z@
2254 \eql@widthdata@for{%
2255 \ifdim\eql@tagwidth@>>z@
2256 \eql@widthdata@calc
2257 \eql@adjust@columns@test
2258 \fi
2259 }%
2260 \else
2261 \ifnum\eql@numbering@target@>>z@
2262 \ifnum\eql@numbering@target@>\eql@totalrows@<\else
2263 \eql@widthdata@get\eql@numbering@target@
2264 \eql@tagwidth@\eql@tagwidth@single@
2265 \eql@widthdata@calc
2266 \eql@adjust@columns@test
2267 \fi
2268 \fi
2269 \fi

```

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

```

2270 \advance\eql@totalwidth@<\dimexpr
2271 \eql@columns@inter@<\eql@colsep@+<\eql@marginleft@<\relax

```

2272 }

## Placement for Right Tags.

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

2273 `\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?

2274 `\@tempdima\dimexpr\displaywidth-\eq@line@width@-\eq@tagwidth@relax`

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

2275 `\ifdim\@tempdima<\dimexpr`

2276 `\eq@marginleft@+\eq@line@widthsep@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).

2277 `\ifdim\@tempdima<\dimexpr`

2278 `\eq@marginleft@min@+\eq@line@widthsep@eq@colsepmin@relax\else`

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

2279 `\eq@adjust@columns@modify@tagsright`

2280 `\fi`

2281 `\fi`

2282 }

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

2283 `\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.

2284 `\ifnum\eq@line@widthsep@>\z@`

2285 `\dimen@ \@tempdima`

2286 `\count@ \eq@line@widthsep@`

2287 `\ifdefined\eq@layoutleft`

2288 `\advance\dimen@-\eq@marginleft@`

2289 `\else`

2290 `\ifdefined\eq@columns@fulllength\else`

2291 `\advance\count@\@ne`

2292 `\fi`

2293 `\fi`

2294 `\divide\dimen@\count@`

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

2295 `\ifdim\dimen@<\eq@colsep@`

2296 `\ifdim\dimen@<\eq@colsepmin@`

2297 `\eq@colsep@\eq@colsepmin@`

2298 `\else`

```

2299      \eq@colsep@ \dimen@
2300      \fi
2301      \fi
2302      \fi

```

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

```

2303      \dimen@ \dimexpr \@tempdima - \eq@line@widthsep@ \eq@colsep@ \relax
2304      \ifdim \eq@marginleft@ > \dimen@
2305          \eq@marginleft@ \dimen@
2306      \fi
2307 }

```

### Placement for Left Tags.

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

```

2308 \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.

```

2309      \count@ \numexpr \eq@columns@inter@ - \eq@line@availsep@ \relax
2310      \@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.

```

2311      \ifdim \@tempdima > \dimexpr
2312          \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).

```

2313      \ifdim \eq@tagwidth@ <
2314          \ifdefined \eq@layoutleft
2315              \glueexpr \eq@layoutleftmarginmax \relax
2316          \else
2317              \displaywidth
2318          \fi

```

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

```

2319          \ifdim \@tempdima > \dimexpr
2320              \displaywidth - \eq@totalwidth@ - \count@ \eq@colsepmin@ \relax \else

```

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

```

2321          \eq@adjust@columns@modify@tagsleft
2322      \fi
2323      \fi
2324      \fi
2325 }

```

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

```

2326 \def \eq@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.

```

2327 \ifnum\count@>\z@
2328   \dimen@ \dimexpr \displaywidth - \eq@totalwidth - \@tempdima \relax
2329   \ifdefined\eq@columns@fulllength\else
2330     \advance\count@\@ne
2331   \fi
2332   \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.

```

2333   \ifdim\dimen@<\eq@colsep@
2334     \ifdim\dimen@<\eq@colsepmin@
2335       \dimen@\eq@colsepmin@
2336     \fi
2337     \advance\dimen@-\eq@colsep@
2338     \advance\eq@marginleft@-\eq@columns@inter@\dimen@
2339     \advance\eq@colsep@\dimen@
2340   \fi
2341 \fi

```

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

```

2342 \dimen@\dimexpr \@tempdima - \eq@line@availsep - \eq@colsep@ \relax
2343 \ifdim\eq@marginleft@<\dimen@
2344   \eq@marginleft@\dimen@
2345 \fi
2346 }

```

## 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:

```

2347 \ifdefined\inf@bad\else%
2348   \newcount\inf@bad
2349   \inf@bad1000000\relax
2350 \fi

```

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

```

2351 \let\eq@restore@hfuzz\@empty
2352 \def\eq@save@hfuzz{\edef\eq@restore@hfuzz{\hfuzz\the\hfuzz\relax}}

```

`\eq@alignbadness@` The registers `\eq@alignbadness@` and `\eq@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:

```

2353 \newcount\eq@alignbadness@
2354 \newcount\eq@tagbadness@

```

```

2355 \newcount\eql@arrange@badness@
2356 \eql@alignbadness@\inf@bad
2357 \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@`:

```

2358 \def\eql@arrange@try#1{%
2359   \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax
2360     \setbox\eql@cellbox@\hbox to\displaywidth{%
2361       \unhbox\eql@cellbox@\unkern\kern#1}%
2362     \eql@arrange@badness@\badness
2363   \else
2364     \eql@arrange@badness@\m@ne
2365   \fi
2366 }

```

**\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`:

```

2367 \def\eql@arrange@print#1#2{%
2368   \eql@restore@hfuzz
2369   \hbox to\displaywidth{%
2370     #1%
2371     \unhbox\eql@cellbox@\unkern
2372     #2%
2373     \eql@tagging@mathaddlast
2374   }%
2375 }

```

**\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:

```

2376 \def\eql@arrange@print@alignleft#1#2{%
2377   \ifnum\eql@row@=\@ne
2378     \global\eql@display@firstavail@#1%
2379   \fi
2380   \eql@arrange@print{\kern#1}{\kern#2}%
2381 }

2382 \def\eql@arrange@print@alignright#1#2{%
2383   \ifnum\eql@row@=\@ne
2384     \global\eql@display@firstavail@\dimexpr
2385       \displaywidth-\eql@cellwidth@-#2\relax
2386   \fi
2387   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
2388 }

2389 \def\eql@arrange@print@aligncenter#1{%
2390   \ifnum\eql@row@=\@ne
2391     \global\eql@display@firstavail@\dimexpr
2392       (\displaywidth-\eql@cellwidth@+#1)/2\relax
2393   \fi
2394   \ifdim#1>\z@

```

```

2395 \eql@arrange@print{\kern#1\hfil}{}%
2396 \else
2397 \eql@arrange@print{\hfil}{\kern-#1}%
2398 \fi
2399 }

```

`\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:

```

2400 \def\eql@arrange@init{%
2401 \eql@save@hfuzz
2402 \hfuzz\maxdimen
2403 \eql@shape@select
2404 }

```

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

```

2405 \def\eql@arrange@print@tag{%
2406 \eql@tagging@tagaddbox
2407 \csname eql@arrange%
2408 @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
2409 @init\endcsname
2410 \csname eql@arrange%
2411 @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
2412 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
2413 \endcsname
2414 }

2415 \def\eql@arrange@print@notag{%
2416 \eql@tagging@tagaddbox
2417 \csname eql@arrange%
2418 @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
2419 @init\endcsname
2420 \csname eql@arrange%
2421 @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
2422 @notag\endcsname
2423 }

```

### J.3 Central Alignment

**TODO:** describe

```

2424 \def\eql@arrange@aligncenter@init{%
2425 \eql@tagging@aligncenter
2426 \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
2427 +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
2428 }

```

**TODO:** describe

```

2429 \def\eql@arrange@aligncenter@notag{%
2430 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>
2431 \dimexpr\eql@line@offset@<\eql@marginleft@min@
2432 \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
2433 \else
2434 \eql@line@offset@
2435 \fi

```

```

2436 \eqL@arrange@print@aligncenter\eqL@line@offset@
2437 \else
2438 \ifdim\eqL@line@offset@<\eqL@marginleft@min@
2439 \eqL@arrange@print@alignleft\eqL@marginleft@min@\z@
2440 \else
2441 \eqL@arrange@print@alignright\eqL@marginleft@min@\z@
2442 \fi
2443 \fi
2444 }

```

**TODO:** describe

```

2445 \def\eqL@arrange@aligncenter@tagsright{%
2446 \ifdim\dimexpr\displaywidth-\eqL@cellwidth@>\relax
2447 \ifdim\eqL@line@offset@<\dimexpr\eqL@marginleft@min@-\eqL@tagwidth@>\relax
2448 \dimexpr\tw@\eqL@marginleft@min@-\eqL@line@offset@>\relax
2449 \else
2450 \dimexpr\tw@\eqL@tagwidth@+\eqL@line@offset@>\relax
2451 \fi
2452 \eqL@arrange@print@aligncenter\eqL@line@offset@
2453 \eqL@tagbox@print@tagsright
2454 \else
2455 \eqL@arrange@try{\dimexpr\eqL@tagwidth@+\eqL@marginleft@min@>\relax}%
2456 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
2457 \ifdim\eqL@line@offset@<\dimexpr\eqL@marginleft@min@-\eqL@tagwidth@>\relax
2458 \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
2459 \else
2460 \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
2461 \fi
2462 \eqL@tagbox@print@tagsright
2463 \else
2464 \eqL@arrange@aligncenter@notag
2465 \eqL@tagbox@print@tagsright@raise
2466 \fi
2467 \fi
2468 }

```

```

2469 \def\eqL@arrange@aligncenter@tagsleft{%
2470 \ifdim\eqL@tagwidth@>\eqL@marginleft@min@
2471 \ifdim\dimexpr\displaywidth-\eqL@cellwidth@>\relax
2472 \ifdim\eqL@line@offset@<\eqL@tagwidth@
2473 \dimexpr\tw@\eqL@tagwidth@-\eqL@line@offset@>\relax
2474 \else
2475 \eqL@line@offset@
2476 \fi
2477 \eqL@tagbox@print@tagsleft
2478 \eqL@arrange@print@aligncenter\eqL@line@offset@
2479 \else
2480 \eqL@arrange@try\eqL@tagwidth@
2481 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
2482 \eqL@tagbox@print@tagsleft
2483 \ifdim\eqL@line@offset@<\eqL@tagwidth@
2484 \eqL@arrange@print@alignleft\eqL@tagwidth@\z@
2485 \else
2486 \eqL@arrange@print@alignright\eqL@tagwidth@\z@
2487 \fi
2488 \else
2489 \eqL@tagbox@print@tagsleft@raise
2490 \eqL@arrange@aligncenter@notag

```

```

2491     \fi
2492     \fi
2493   \else
2494     \eq\tagbox@print@tagsleft
2495     \eq\arrange@aligncenter@notag
2496   \fi
2497   \eq\display@firstavail@set\z@
2498 }

```

## J.4 Left Alignment

```

2499 \def\eq\arrange@alignleft@init{%
2500   \eq\tagging@alignleft
2501   \eq\line@offset@\dimexpr\eq\marginleft@+\eq\shape@amount@\relax
2502   \ifdim\eq\line@offset@<\eq\marginleft@min@
2503     \eq\line@offset@\eq\marginleft@min@
2504   \fi
2505 }

2506 \def\eq\arrange@alignleft@notag{%
2507   \ifdim\eq\line@offset@>\eq\marginleft@min@
2508     \eq\arrange@try\eq\line@offset@
2509     \ifnum\eq\arrange@badness@<\eq\alignbadness@
2510       \eq\arrange@print@alignleft\eq\line@offset@\z@
2511     \else
2512       \eq\arrange@print@alignright\eq\marginleft@min@\z@
2513     \fi
2514   \else
2515     \eq\arrange@print@alignleft\eq\marginleft@min@\z@
2516   \fi
2517 }

2518 \def\eq\arrange@alignleft@tagsright{%
2519   \eq\arrange@try{\dimexpr\eq\line@offset@+\eq\tagwidth@\relax}%
2520   \ifnum\eq\arrange@badness@<\eq\alignbadness@
2521     \eq\arrange@print@alignleft\eq\line@offset@\eq\tagwidth@
2522     \eq\tagbox@print@tagsright
2523   \else
2524     \ifdim\eq\line@offset@>\eq\marginleft@min@
2525       \eq\arrange@try{\dimexpr\eq\marginleft@min@+\eq\tagwidth@\relax}%
2526     \fi
2527     \ifnum\eq\arrange@badness@<\eq\tagbadness@
2528       \eq\arrange@print@alignright\eq\marginleft@min@\eq\tagwidth@
2529       \eq\tagbox@print@tagsright
2530     \else
2531       \eq\arrange@alignleft@notag
2532       \eq\tagbox@print@tagsright@raise
2533     \fi
2534   \fi
2535 }

2536 \def\eq\arrange@alignleft@tagsleft{%
2537   \ifdim\eq\tagwidth@>\eq\marginleft@min@
2538     \ifdim\eq\line@offset@>\eq\tagwidth@
2539       \eq\arrange@try\eq\line@offset@
2540       \ifnum\eq\arrange@badness@<\eq\alignbadness@
2541         \eq\tagbox@print@tagsleft
2542         \eq\arrange@print@alignleft\eq\line@offset@\z@
2543       \else
2544         \eq\arrange@try\eq\tagwidth@

```



```

2545         \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
2546         \eqL@tagbox@print@tagsleft
2547         \eqL@arrange@print@alignright\eqL@tagwidth@\z@
2548     \else
2549         \eqL@tagbox@print@tagsleft@raise
2550         \eqL@arrange@print@alignright\eqL@marginleft@min@\z@
2551     \fi
2552 \fi
2553 \else
2554     \eqL@arrange@try\eqL@tagwidth@
2555     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
2556     \eqL@tagbox@print@tagsleft
2557     \eqL@arrange@print@alignleft\eqL@tagwidth@\z@
2558 \else
2559     \eqL@tagbox@print@tagsleft@raise
2560     \eqL@arrange@alignleft@notag
2561 \fi
2562 \fi
2563 \else
2564     \eqL@tagbox@print@tagsleft
2565     \eqL@arrange@alignleft@notag
2566 \fi
2567 \eqL@display@firstavail@set\z@
2568 }

```

## J.5 Right Alignment

```

2569 \def\eqL@arrange@alignright@init{%
2570     \eqL@tagging@alignright
2571     \eqL@line@offset@\dimexpr\eqL@marginright@-\eqL@shape@amount@\relax
2572     \ifdim\eqL@line@offset@<\z@
2573     \eqL@line@offset@\z@
2574 \fi
2575 }

```

**TODO:** describe

```

2576 \def\eqL@arrange@alignright@notag{%
2577     \ifdim\eqL@line@offset@>\z@
2578     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@\relax}%
2579     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
2580     \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
2581     \else
2582     \eqL@arrange@print@alignleft\eqL@marginleft@min@\z@
2583 \fi
2584 \else
2585     \eqL@arrange@print@alignright\eqL@marginleft@min@\z@
2586 \fi
2587 }

```

**TODO:** describe

```

2588 \def\eqL@arrange@alignright@tagsright{%
2589     \ifdim\eqL@line@offset@>\eqL@tagwidth@
2590     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@\relax}%
2591     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
2592     \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
2593     \eqL@tagbox@print@tagsright
2594 \else
2595     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@\relax}%
2596     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@

```

```

2597         \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
2598         \eql@tagbox@print@tagsright
2599     \else
2600         \eql@arrange@print@alignleft\eql@marginleft@min@\z@
2601         \eql@tagbox@print@tagsright@raise
2602     \fi
2603 \fi
2604 \else
2605     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
2606     \ifnum\eql@arrange@badness@<\eql@tagbadness@
2607         \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
2608         \eql@tagbox@print@tagsright
2609     \else
2610         \eql@arrange@alignright@notag
2611         \eql@tagbox@print@tagsright@raise
2612     \fi
2613 \fi
2614 }

```

**TODO:** describe

```

2615 \def\eql@arrange@alignright@tagsleft{%
2616     \ifdim\eql@tagwidth@>\eql@marginleft@min@
2617         \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
2618         \ifnum\eql@arrange@badness@<\eql@alignbadness@
2619             \eql@tagbox@print@tagsleft
2620             \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
2621         \else
2622             \ifdim\eql@line@offset@>\z@
2623                 \eql@arrange@try\eql@tagwidth@
2624             \fi
2625             \ifnum\eql@arrange@badness@<\eql@tagbadness@
2626                 \eql@tagbox@print@tagsleft
2627                 \eql@arrange@print@alignleft\eql@tagwidth@\z@
2628             \else
2629                 \eql@tagbox@print@tagsleft@raise
2630                 \eql@arrange@alignright@notag
2631             \fi
2632         \fi
2633     \else
2634         \eql@tagbox@print@tagsleft
2635         \eql@arrange@alignright@notag
2636     \fi
2637     \eql@display@firstavail@set\z@
2638 }

```

## K Equations Box Environment

**TODO:** describe

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

**TODO:** vspace?!

### K.1 Line Breaks

`\eql@box@cr`

```

2639 \protected\def\eql@box@cr{%

```

```

2640 \eql@ampprotecttwo{\eql@ifnextchar@tight[]\eql@box@cr@skip\eql@box@cr@
2641 }
2642 \def\eql@box@cr@{%
2643 \eql@punct@apply@line
2644 \eql@hook@lineout
2645 \eql@box@lastcell
2646 \cr
2647 }
2648 \def\eql@box@cr@skip[#1]{%
2649 \eql@box@cr@
2650 \noalign{%
2651 \vskip\glueexpr#1\relax
2652 }%
2653 }

```

## K.2 Stacked Mode

```

2654 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep}
2655 \def\eql@box@open@stacked{%
2656 % \TODO templates
2657 \eql@shape@align@enable
2658 \let\eql@box@lastcell\eql@box@lastcell@stacked
2659 \everycr{\noalign{%
2660 (dev)\eql@dev{starting line \the\eql@row@}%
2661 \global\advance\eql@row@\@ne
2662 }}%
2663 \tabskip\z@skip
2664 \halign\bgroup
2665 &%
2666 \eql@shape@pos@\m@ne
2667 \setbox\eql@cellbox@\hbox{%
2668 \eql@strut@cell
2669 \@lign
2670 $\m@th\displaystyle
2671 \eql@hook@colin
2672 ##%
2673 \eql@punct@apply@col
2674 \eql@hook@colout
2675 \eql@tagging@mathsave
2676 $%
2677 \eql@tagging@mathaddlast
2678 }%
2679 \ifdefined\eql@frame@cmd
2680 \eql@frame@print
2681 \fi
2682 \ifnum\eql@shape@pos@<\z@
2683 \ifcase-\eql@shape@pos@
2684 \or
2685 \eql@shape@eval
2686 \or
2687 \@tempdima\eql@shape@amount@
2688 \eql@shape@eval
2689 \eql@shape@amount@\@tempdima
2690 \or
2691 \@tempdima\eql@shape@amount@
2692 \eql@shape@eval
2693 \advance\eql@shape@amount@\@tempdima
2694 \fi

```

```

2695     \fi
2696     \ifcase\eq@shape@pos@
2697         \kern\eq@shape@amount@
2698         \box\eq@cellbox@
2699         \hskip\glueexpr\eq@paddingleft@+\eq@paddingright@
2700         -\eq@shape@amount@+\@flushglue\relax
2701         \eq@tagging@alignleft
2702     \or
2703         \hskip\glueexpr\eq@paddingleft@+\eq@shape@amount@+\@flushglue\relax
2704         \box\eq@cellbox@
2705         \hskip\glueexpr\eq@paddingright@-\eq@shape@amount@+\@flushglue\relax
2706         \eq@tagging@aligncenter
2707     \or
2708         \hskip\glueexpr\eq@paddingleft@+\eq@paddingright@
2709         +\eq@shape@amount@+\@flushglue\relax
2710         \box\eq@cellbox@
2711         \kern-\eq@shape@amount@
2712         \eq@tagging@alignright
2713     \fi
2714     \tabskip\eq@colsep@\relax
2715 \crrc
2716 \noalign{%
2717     \eq@hook@blockbefore
2718 }%
2719 \eq@hook@blockin
2720 }
2721 \def\eq@mode@stacked{\let\eq@box@open\eq@box@open@stacked}

```

### K.3 Aligned Mode

```

2722 \def\eq@box@lastcell@odd{%
2723     &\omit
2724     \ifdefined\eq@frame@cmd
2725         \eq@frame@measure
2726         \advance\eq@prevwidth@\eq@frame@margin@
2727         \eq@frame@print
2728     \fi
2729     \kern-\eq@prevwidth@
2730     \unhbox\eq@cellbox@
2731     \hfil
2732     &\omit\kern-\eq@colsep@
2733 }%
2734 \def\eq@box@lastcell@even{&\omit\kern-\eq@colsep@}
2735 \def\eq@box@open@aligned{%
2736 % \TODO templates
2737 \eq@shape@align@disable
2738 \let\eq@box@lastcell\@empty
2739 \everycr{\noalign{%
2740 (dev)\eq@dev{starting new line}%
2741 }}%
2742 \tabskip\z@skip
2743 \halign\bgroup
2744     &%
2745     \let\eq@box@lastcell\eq@box@lastcell@odd
2746     \global\setbox\eq@cellbox@\hbox{%
2747         \eq@strut@cell
2748         \@lign
2749         $\m@th\displaystyle

```

```

2750         \eql@hook@colin
2751         ##%
2752         \eql@class@innerleft
2753         \eql@hook@innerleft
2754         \eql@tagging@mathsave
2755         $%
2756         \eql@tagging@mathaddlast
2757     }%
2758     \global\eql@prevwidth@\wd\eql@cellbox@
2759     \hfil
2760     \kern\wd\eql@cellbox@
2761     \ifdefined\eql@frame@cmd
2762         \eql@frame@measure
2763         \kern\eql@frame@margin@
2764     \fi
2765     \tabskip\z@skip
2766     &%
2767     \let\eql@box@lastcell\eql@box@lastcell@even
2768     \setbox\eql@cellbox@\hbox{%
2769         \unhbox\eql@cellbox@
2770         \eql@strut@cell
2771         \@lign
2772         $\m@th\displaystyle
2773         \eql@hook@innerright
2774         \eql@class@innerright@sel
2775         ##%
2776         \eql@punct@apply@col
2777         \eql@hook@colout
2778         \eql@tagging@mathsave
2779         $%
2780         \eql@tagging@mathaddlast
2781     }%
2782     \ifdefined\eql@frame@cmd
2783         \eql@frame@measure
2784         \advance\eql@prevwidth@\eql@frame@margin@
2785         \eql@frame@print
2786     \fi
2787     \kern-\eql@prevwidth@
2788     \unhbox\eql@cellbox@
2789     \hfil
2790     \tabskip\eql@colsep@\relax
2791     \crrr
2792     \noalign{%
2793         \eql@hook@blockbefore
2794     }%
2795     \eql@hook@blockin
2796 }
2797 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

## K.4 Main

```

2798 \let\eql@box@box\vcenter
2799 \let\eql@box@open\@undefined
2800 \let\eql@box@frame\@firstofone
2801 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}
2802 \def\eql@box@close{%
2803     \ifvmode\else
2804         \global\eql@totalrows@\eql@row@

```

```

2805     \eql@punct@apply@block
2806     \eql@box@cr@
2807     \fi
2808     \noalign{%
2809         \eql@hook@blockafter
2810     }%
2811     \eql@tagging@tablesaveinner
2812 \egroup
2813 }

```

`\eql@box@vcenter`

```

2814 \def\eql@box@vcenter#1{%
2815     \ifmmode
2816         \vcenter{#1}%
2817     \else
2818         $\m@th\vcenter{#1}$%
2819     \fi
2820 }

```

`\eql@box@start`

```

2821 \let\eql@box@endmath\eql@false
2822 \def\eql@box@start{%
2823     \relax
2824     \ifmmode
2825         \let\eql@box@endmath\eql@false
2826     \else
2827         \let\eql@box@endmath\eql@true
2828         \expandafter$%$
2829     \fi
2830     \eql@box@processopt
2831     \eql@stack@save@boxed
2832     \let\eql@layoutleft\eql@false
2833     \eql@row@z@
2834     \eql@totalrows@ \@M
2835     \eql@shape@select
2836     \setbox\z@\ifx\eql@box@box\vcenter
2837         \expandafter\vbox
2838     \else
2839         \expandafter\eql@box@box
2840     \fi\bgroup
2841     \eql@display@nest
2842     \let\\ \eql@box@cr
2843     \eql@spread@set
2844     \eql@strut@make
2845     \eql@box@open
2846 }

```

`\eql@box@end`

```

2847 \def\eql@box@end{%
2848     \eql@box@close
2849     \egroup
2850     \eql@box@frame{%
2851         \ifdefined\eql@display@marginleft
2852             \hskip\glueexpr\eql@display@marginleft\relax
2853         \fi
2854         \ifx\eql@box@box\vcenter

```

```

2855     \eql@box@vcenter{\unvbox\z@}%
2856   \else
2857     \box\z@
2858   \fi
2859   \eql@tagging@tableaddinner
2860   \ifdefined\eql@display@marginright
2861     \hskip\glueexpr\eql@display@marginright\relax
2862   \fi
2863 }%
2864 \eql@stack@restore
2865 \ifdefined\eql@box@endmath
2866   \expandafter$%$
2867 \fi
2868 }

```

## K.5 Environment

`equationsbox` (*env.*)

```

2869 \newenvironment{equationsbox}{%
2870 (dev)\eql@dev@enterenv
2871   \eql@ampprotect\eql@box@testall\eql@box@start
2872 }{%
2873   \eql@box@end
2874 (dev)\eql@dev@leaveenv
2875 }

```

```

2876 \def\eql@box@testall{\eql@box@testtilde}
2877 \def\eql@box@testtilde#1{%
2878   \eql@ifnextgobble@tight~%
2879   {\eqnaddopt{lines}\eql@box@testopt{#1}}%
2880   {\eql@box@testopt{#1}}}
2881 \def\eql@box@testopt#1{%
2882   \eql@ifnextchar@tight[%]
2883   {\eql@box@adopt{#1}}%
2884   {#1}}
2885 \def\eql@box@adopt#1[#2]{\eqnaddopt{#2}#1}

```

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

```

2886 \def\eql@box@processopt{%
2887   \eql@frame@reset
2888   \let\eql@box@frame\@firstofone
2889   \let\eql@display@marginleft\@undefined
2890   \let\eql@display@marginright\@undefined
2891   \eql@nextopt@process{equationsbox}%
2892   \let\eql@punct@block\eql@punct@main
2893   \let\eql@punct@main\relax
2894   \eql@colsep@\glueexpr\eql@box@colsep\relax
2895   \ifdefined\eql@paddingleft@val
2896     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
2897   \else
2898     \eql@paddingleft@\z@
2899   \fi
2900   \ifdefined\eql@paddingright@val
2901     \eql@paddingright@\glueexpr\eql@paddingright@val\relax
2902   \else
2903     \eql@paddingright@\z@

```

```

2904 \fi
2905 \eql@indent@\glueexpr\eql@indent@val\relax
2906 }

```

## L Single-Line Equation

**TODO:** describe

### L.1 Native Mode

```

2907 \def\eql@single@start@native{%
2908   \eql@display@init
2909   \eql@display@print
2910   \let\raisetag\eql@raisetag@default
2911   \eql@shape@align@disable
2912   \eql@hook@eqin
2913 %   \mathopen{}%
2914 }%

```

**TODO:** describe

```

2915 \def\eql@single@end@native{%
2916 %   \mathclose{}%
2917   \if@eqnsw
2918     \ifdefined\eql@tagsleft
2919       \leqno
2920     \else
2921       \eqno
2922     \fi
2923     \eql@compose@print
2924   \fi
2925   \eql@display@penalty
2926   \eql@display@vspace@native
2927 }%

```

### L.2 Print

```

2928 \def\eql@single@start@print{%
2929   \eql@display@init
2930   \eql@display@print
2931   \eql@shape@align@enable
2932   \eql@totalrows@\@ne
2933   \eql@row@\@ne
2934   \eql@arrange@init
2935   \eql@shape@eval
2936   \prevgraf\numexpr\prevgraf+\@ne\relax
2937   \setbox\eql@cellbox@\hbox\bgroup
2938     \eql@restore@hfuzz
2939     \eql@strut@cell
2940     $\m@th\displaystyle%$
2941     \eql@hook@eqin
2942 }
2943 \def\eql@single@end@print{%
2944   \eql@tagging@mathsave
2945   $%$
2946   \hfil

```



```

2947 \kern\z@
2948 \egroup
2949 \prevgraf\numexpr\prevgraf-\@ne\relax
2950 \ifdefined\eql@frame@cmd
2951 \eql@frame@adjust
2952 \fi
2953 \eql@cellwidth@\wd\eql@cellbox@
2954 \eql@line@height@\ht\eql@cellbox@
2955 \eql@line@depth@\dp\eql@cellbox@
2956 \eql@totalwidth@\eql@cellwidth@
2957 \if@eqnsw
2958 \eql@tagbox@make\eql@compose@print
2959 \eql@taggrows@\@ne
2960 \else
2961 \eql@tagwidth@\z@
2962 \eql@taggrows@\z@
2963 \fi
2964 \eql@tagwidth@max@\eql@tagwidth@
2965 \eql@columns@inter@\z@
2966 \eql@adjust@calc@lines
2967 \eql@display@halign@init{}%
2968 \halign{##\crr
2969 \noalign{\eql@display@halign@start}%
2970 \if@eqnsw
2971 \eql@arrange@print@tag
2972 \else
2973 \eql@arrange@print@notag
2974 \fi
2975 \cr
2976 \noalign{\eql@display@halign@end}%
2977 \eql@tagging@tablesavelines
2978 }%
2979 \eql@display@close
2980 }

```

## M Multi-Line with Single Column

### M.1 Measure

**TODO:** describe

```

2981 \def\eql@lines@measure@line@begin{%
2982 (dev)\eql@dev{starting line \the\eql@row@}%
2983 \eql@numbering@measure@line@begin
2984 \eql@hook@linein
2985 }

```

**TODO:** describe

```

2986 \def\eql@lines@measure@line@end{%
2987 \eql@punct@apply@line
2988 \eql@hook@lineout
2989 }

```

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

```

2990 \def\eql@lines@measure@cell{%
2991   \ifdefined\eql@frame@cmd
2992     \eql@frame@print
2993   \fi
2994   \eql@cellwidth@\wd\eql@cellbox@
2995   \eql@widthdata@startrow
2996   \eql@widthdata@savcell
2997   \kern\eql@cellwidth@
2998 }

```

**TODO:** describe

```

2999 \def\eql@lines@measure@tag{%
3000   \eql@tagwidth@\z@
3001   \ifnum\eql@numbering@target@<\z@
3002     \if@eqnsw
3003       \eql@tagbox@make\eql@compose@measure
3004     \fi
3005   \fi
3006   \eql@widthdata@savetag
3007 }

```

**\eql@lines@measure**

```

3008 \def\eql@lines@measure{%
3009   (dev)\eql@dev@enter\eql@lines@measure
3010   \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3011   \eql@totalrows@\@M
3012   \eql@shape@select

3013   \setbox\z@\vbox{\halign{%
3014     \eql@shape@eval
3015     \setbox\eql@cellbox@\hbox{%
3016       \@lign
3017       $\m@th\displaystyle
3018       \eql@hook@colin
3019       ##%
3020       \eql@punct@apply@col
3021       \eql@hook@colout
3022       $%
3023     }%
3024     \eql@lines@measure@cell
3025     \eql@lines@measure@tag
3026   \crcr

3027   \noalign{%
3028     \eql@hook@blockbefore
3029   }%
3030   \eql@hook@blockin
3031   \eql@scan@body
3032   \ifvmode\else
3033     \eql@totalrows@\eql@row@
3034     \eql@punct@apply@block
3035     \eql@hook@blockout
3036     \eql@display@endline
3037   \cr
3038   \fi
3039   \omit
3040   \cr
3041   \noalign{%

```

```

3042     \eql@hook@blockafter
3043   }%
3044 }}%

3045 \eql@measure@close

3046 \setbox\z@\vbox{%
3047   \unvbox\z@
3048   \unpenalty
3049   \global\setbox\@ne\lastbox
3050 }%
3051 \eql@totalwidth@\wd\@ne

3052 (dev)\eql@dev@leave\eql@lines@measure
3053 }

```

## M.2 Column Placement

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

```

3054 \def\eql@lines@adjust{%
3055   \eql@adjust@calc@lines
3056   \eql@numbering@best@eval
3057 }

```

## M.3 Print

**TODO:** describe

nes@print@line@begin

```

3058 \def\eql@lines@print@line@begin{%
3059 (dev)\eql@dev{starting line \the\eql@row}%
3060 \eql@numbering@print@line@begin
3061 \eql@hook@linein
3062 }

```

**TODO:** describe

```

3063 \def\eql@lines@print@line@end{%
3064   \eql@punct@apply@line
3065   \eql@hook@lineout
3066 }

```

**TODO:** describe

```

3067 \def\eql@lines@print@line@adjust{%
3068   \ifdefined\eql@frame@cmd
3069     \eql@frame@adjust
3070   \fi
3071   \eql@numbering@print@line@eval
3072   \eql@cellwidth@\wd\eql@cellbox@
3073   \eql@line@height@\ht\eql@cellbox@
3074   \eql@line@depth@\dp\eql@cellbox@
3075   \if@eqnsw
3076     \eql@tagbox@make\eql@compose@print
3077     \eql@arrange@print@tag
3078   \else
3079     \eql@arrange@print@notag

```

```

3080 \fi
3081 }

```

**TODO:** describe

```

3082 \def\eq@lines@print{%
3083 (dev)\eq@dev@center\eq@lines@print
3084 \eq@arrange@init
3085 \eq@display@halign@init\eq@lines@print@line@begin
3086 \eq@display@halign@letcr\eq@lines@print@line@end
3087 \tabskip\z@skip

3088 \halign{%
3089 \eq@shape@eval
3090 \setbox\eq@cellbox@\hbox{%
3091 \eq@restore@hfuzz
3092 \eq@strut@cell
3093 \@lign
3094 $\m@th\displaystyle
3095 \eq@hook@colin
3096 ##%
3097 \eq@punct@apply@col
3098 \eq@hook@colout
3099 \eq@tagging@mathsave
3100 $%
3101 \hfil
3102 \kern\z@
3103 }%
3104 \eq@lines@print@line@adjust
3105 \cr

3106 \noalign{%
3107 \eq@display@halign@start
3108 \eq@numbering@print@block@begin
3109 \eq@hook@blockbefore
3110 }%
3111 \eq@hook@blockin
3112 \eq@scan@body
3113 \ifvmode\else
3114 \relax
3115 \eq@punct@apply@block
3116 \eq@hook@blockout
3117 \eq@display@endline
3118 \cr
3119 \fi
3120 \noalign{%
3121 \eq@hook@blockafter
3122 \eq@display@halign@end
3123 (dev)\eq@dev@leave\eq@lines@print
3124 }%
3125 \eq@tagging@tablesavelines
3126 }%
3127 }

```

## N Multi-Line with Multiple Columns

**TODO:** describe

## N.1 Support

**TODO:** describe

```
\eql@columns@add@amp
@columns@completerow
3128 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3129 \def\eql@columns@completerow{%
3130   \count@\eql@totalcolumns@
3131   \advance\count@\@ne
3132   \advance\count@-\eql@column@
3133   \edef\eql@tmp{%
3134     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3135   \eql@tmp
3136 }

3137 \def\eql@columns@overfull{%
3138   \dimen@\eql@line@width@
3139   \advance\dimen@-\hfuzz
3140   \ifdim\dimen@>\displaywidth
3141     \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3142     \wd\z@\z@
3143     \ht\z@\eql@line@height@
3144     \dp\z@\eql@line@depth@
3145     \box\z@
3146   \fi
3147 }
```

## N.2 Measure

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

```
s@measure@line@begin

3148 \def\eql@columns@measure@line@begin{%
3149 (dev)\eql@dev{starting line \the\eql@row@}%
3150   \global\eql@column@\z@
3151   \eql@numbering@measure@line@begin
3152   \eql@hook@linein
3153 }

3154 \def\eql@columns@measure@cell{%
3155   \eql@cellwidth@\wd\eql@cellbox@
3156   \ifdefined\eql@frame@cmd
3157     \eql@frame@measure
3158     \advance\eql@cellwidth@\eql@frame@margin@
3159   \ifodd\eql@column@\else
3160     \eql@frame@reset
3161   \fi
3162 \fi
3163 \ifnum\eql@column@=\@ne
3164   \eql@widthdata@startrow
3165 \fi
3166 \ifodd\eql@column@
3167   \eql@shape@pos@\tw@
3168 \else
3169   \eql@shape@pos@\z@
3170 \fi
```

```

3171 \eq\shape@amount@\z@
3172 \eq\widthdata@savcell
3173 \ifodd\eq\column@\else
3174 \eq\widthdata@savesep
3175 \fi
3176 \kern\eq\cellwidth@
3177 }

```

mns@measure@line@end

```

3178 \def\eq\columns@measure@line@end{%
3179 \eq\punct@apply@line
3180 \eq\hook@lineout
3181 &\omit
3182 \ifnum\eq\column@>\eq\totalcolumns@
3183 \global\eq\totalcolumns@\eq\column@
3184 \fi

```

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

```

3185 \ifdefined\eq\frame@cmd
3186 \advance\eq\column@\@ne
3187 \wd\eq\cellbox@\z@
3188 \eq\columns@measure@cell
3189 \fi
3190 \eq\columns@measure@tag
3191 }

```

@columns@measure@tag

```

3192 \def\eq\columns@measure@tag{%
3193 \eq\tagwidth@\z@
3194 \ifnum\eq\numbering@target@<\z@
3195 \if@eqnsw
3196 \eq\tagbox@make\eq\compose@measure
3197 \fi
3198 \fi
3199 \eq\widthdata@savetag
3200 }

```

\eq\columns@measure

```

3201 \def\eq\columns@measure{%
3202 (dev)\eq\dev@enter\eq\columns@measure
3203 \eq\totalcolumns@\z@
3204 \eq\measure@init\eq\columns@measure@line@begin\eq\columns@measure@line@end

3205 \setbox\z@\vbox{\halign{%
3206 &%
3207 \global\advance\eq\column@\@ne
3208 \hfil
3209 \global\setbox\eq\cellbox@\hbox{%
3210 \@lign
3211 $\m@th\displaystyle
3212 \eq\hook@colin
3213 ##%
3214 \eq@class@innerleft

```

```

3215         \eql@hook@innerleft
3216     $%
3217 }%
3218 \global\eql@prevwidth@\wd\eql@cellbox@
3219 \eql@columns@measure@cell
3220 &%
3221 \global\advance\eql@column@\@ne
3222 \setbox\eql@cellbox@\hbox{%
3223     \@lign
3224     $\m@th\displaystyle
3225     \eql@hook@innerright
3226     \eql@class@innerright@sel
3227     ##%
3228     \eql@punct@apply@col
3229     \eql@hook@colout
3230     $%
3231 }%
3232 \eql@columns@measure@cell
3233 \hfil
3234 \crrr

3235 \noalign{%
3236     \eql@hook@blockbefore
3237 }%
3238 \eql@hook@blockin
3239 \eql@scan@body

3240 \ifvmode\else
3241     \eql@punct@apply@block
3242     \eql@hook@blockout
3243     \eql@display@endline
3244     \cr
3245 \fi
3246 \noalign{%
3247     \eql@hook@blockafter
3248 }%

```

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

```

3249     \omit
3250     \eql@column@\@ne
3251     \eql@columns@completerow
3252     \cr
3253 }}%

3254 \eql@measure@close

3255 \setbox\z@\vbox{%
3256     \unvbox\z@
3257     \unpenalty
3258     \global\setbox\@ne\lastbox
3259 }%
3260 \eql@totalwidth@\wd\@ne

```

**TODO:** why not recycle box contents altogether?! **TODO:** maybe add one column of safety?

```

3261 \let\eql@colwidth@tab\@empty
3262 \loop
3263     \setbox\@ne\hbox{%

```

```

3264     \unhbox\@ne
3265     \unskip
3266     \global\setbox\thr@@\lastbox
3267   }%
3268   \ifhbox\thr@@
3269     \eql@colwidth@save{\wd\thr@@}%
3270   \repeat

3271 (dev)\eql@dev@leave\eql@columns@measure
3272 }

```

### N.3 Columns Placement

**TODO:** describe Make sure we have complete pairs of right and left adjusted columns, otherwise add a final empty column:

```

3273 \def\eql@columns@adjust{%
3274   \ifodd\eql@totalcolumns@
3275     \advance\eql@totalcolumns@\@ne
3276   \fi
3277   \eql@adjust@calc@columns
3278 }

```

### N.4 Print

**TODO:** describe

mns@print@line@begin

```

3279 \def\eql@columns@print@line@begin{%
3280 (dev)\eql@dev{starting line \the\eql@row}%
3281   \global\eql@column@\z@
3282   \global\eql@line@pos@\eql@marginleft@
3283   \global\eql@line@width@\z@
3284   \global\eql@line@avail@\eql@totalwidth@
3285   \global\eql@line@height@\z@
3286   \global\eql@line@depth@\z@
3287   \eql@numbering@print@line@begin
3288   \eql@hook@linein
3289 }

```

l@columns@print@cell

```

3290 \def\eql@columns@print@cell{%
3291   \eql@cellwidth@\wd\eql@cellbox@
3292   \ifodd\eql@column@
3293     \ifdefined\eql@frame@cmd
3294       \eql@frame@measure
3295       \advance\eql@cellwidth@\eql@frame@margin@
3296     \fi
3297     \dimen@\z@
3298   \else
3299     \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```

3300   \ifdefined\eql@frame@cmd
3301     \eql@frame@measure

```



```

3302     \advance\eql@cellwidth@\eql@frame@margin@
3303     \advance\eql@prevwidth@\eql@frame@margin@
3304     \eql@frame@print
3305     \fi

```

update height and depth

```

3306     \ifdim\ht\eql@cellbox@>\eql@line@height@
3307         \global\eql@line@height@\ht\eql@cellbox@
3308     \fi
3309     \ifdim\dp\eql@cellbox@>\eql@line@depth@
3310         \global\eql@line@depth@\dp\eql@cellbox@
3311     \fi

```

print box

```

3312     \kern-\eql@prevwidth@
3313     \unhbox\eql@cellbox@
3314     \dimen@-\eql@cellwidth@
3315     \fi

```

enforce given width: hopefully measure was correct, but need a precise width for tag placement

```

3316     \advance\dimen@\eql@colwidth@get\eql@column@\relax
3317     \kern\dimen@

```

update available and used space

```

3318     \dimen@\eql@colwidth@get\eql@column@\relax
3319     \ifdim\eql@cellwidth@>\z@
3320         \ifdim\eql@line@width@=\z@
3321             \eql@line@avail@\eql@line@pos@
3322             \ifodd\eql@column@
3323                 \advance\eql@line@avail@\dimen@
3324                 \advance\eql@line@avail@-\eql@cellwidth@
3325             \fi
3326             \global\eql@line@avail@\eql@line@avail@
3327         \fi
3328         \eql@line@width@\eql@line@pos@
3329         \ifodd\eql@column@
3330             \advance\eql@line@width@\dimen@
3331         \else
3332             \advance\eql@line@width@\eql@cellwidth@
3333         \fi
3334         \global\eql@line@width@\eql@line@width@
3335     \fi
3336     \advance\eql@line@pos@\dimen@
3337     \ifodd\eql@column@\else
3338         \advance\eql@line@pos@\eql@colsep@
3339     \fi
3340     \global\eql@line@pos@\eql@line@pos@
3341 }

```

```

3342 \def\eql@columns@print@trailright{%
3343     &\omit
3344     \global\advance\eql@column@\@ne
3345     \eql@columns@print@cell
3346 }

```

lums@print@line@end

```

3347 \def\eql@columns@print@line@end{%
3348   \eql@punct@apply@line
3349   \eql@hook@lineout
3350 % \TODO add an even column with empty stuff if box processing deferred
3351   \ifodd\eql@column@
3352     \expandafter\eql@columns@print@trailright
3353   \fi
3354   \eql@columns@completerow
3355   \eql@columns@print@tag
3356 }

```

ql@columns@print@tag

```

3357 \def\eql@columns@print@tag{%
3358   \dimen@\eql@totalwidth@
3359   \advance\dimen@\eql@colsep@
3360   \kern-\dimen@

```

determine first line available space

```

3361   \eql@display@firstavail@set\eql@line@avail@
3362   \eql@columns@overfull
3363   \eql@numbering@print@line@eval
3364   \if@eqnsw
3365     \eql@tagbox@make\eql@compose@print
3366     \eql@tagging@tagaddbox
3367     \eql@tagbox@print@cell
3368   \else
3369     \eql@tagging@tagaddbox
3370     \kern\displaywidth
3371   \fi
3372 }

```

\eql@columns@print

```

3373 \def\eql@columns@print{%
3374 (dev)\eql@dev@center\eql@columns@print
3375   \eql@shape@align@disable
3376   \eql@display@halign@init\eql@columns@print@line@begin
3377   \eql@display@halign@letcr\eql@columns@print@line@end
3378   \tabskip\eql@marginleft@

3379   \halign{%
3380     &%
3381     \global\advance\eql@column@\@ne
3382     \hfil
3383     \global\setbox\eql@cellbox@\hbox{%
3384       \eql@strut@cell
3385       \@lign
3386       $\m@th\displaystyle
3387       \eql@hook@colin
3388       ##%
3389       \eql@class@innerleft
3390       \eql@hook@innerleft
3391       \eql@tagging@mathsave
3392       $%
3393       \eql@tagging@mathaddlast
3394     }%
3395     \global\eql@prevwidth@\wd\eql@cellbox@
3396     \eql@columns@print@cell

```

```

3397     \tabskip\z@skip
3398 &%
3399     \global\advance\eql@column@\@ne
3400     \setbox\eql@cellbox@\hbox{%
3401         \unhbox\eql@cellbox@
3402         \eql@strut@cell
3403         \@lign
3404         $\m@th\displaystyle
3405         \eql@hook@innerright
3406         \eql@class@innerright@sel
3407         ##%
3408         \eql@punct@apply@col
3409         \eql@hook@colout
3410         \eql@tagging@mathsave
3411         $%
3412         \eql@tagging@mathaddlast
3413     }%
3414     \eql@columns@print@cell
3415     \hfil
3416     \tabskip\eql@colsep@\relax
3417 \crrr

3418 \noalign{%
3419     \eql@display@halign@start
3420     \eql@numbering@print@block@begin
3421     \eql@hook@blockbefore
3422 }%
3423 \eql@hook@blockin
3424 \eql@scan@body
3425 \ifvmode\else
3426     \relax
3427     \eql@punct@apply@block
3428     \eql@hook@blockout
3429     \eql@display@endline
3430 \cr
3431 \fi
3432 \noalign{%
3433     \eql@hook@blockafter
3434     \eql@display@halign@end
3435 (dev)\eql@dev@leave\eql@columns@print
3436 }%
3437 \eql@tagging@tablesalign
3438 }%
3439 }

```

## O Interface

### O.1 Scanning the Equation Body

The multi-line equatioun 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.

```
3440 \newtoks\eql@scan@reg@
```

`\eql@scan@body@dump` The macro `\eql@scan@body@dump` dumps the equation body from the register so that we  
`\eql@scan@body@rescan` do not have to pass it around in arguments. The macro `\eql@scan@body@rescan` rescans  
`\eql@scan@body` 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:

```
3441 \def\eql@scan@body@dump{\the\eql@scan@reg@}  
3442 \def\eql@scan@body@rescan{%  
3443   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}  
3444 \let\eql@scan@body\eql@scan@body@dump
```

`\eql@scan@addto` We define a macro to append to the token register `\eql@scan@reg@`:

```
3445 \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@`:

```
3446 \def\eql@scan@env#1{%  
3447   (dev)\eql@dev@enter\eql@scan@env  
3448   \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenvir}}%  
3449   \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`:

```
3450   \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%  
3451   \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate  
3452   \ifdefined\eql@scan@par  
3453     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate  
3454   \else  
3455     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar  
3456   \fi  
3457   \eql@scan@iterate  
3458 }
```

`\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:

```
3459 \long\def\eql@scan@env@iterate#1\end#2{%  
3460   \edef\eql@scan@stack{%  
3461     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%  
3462   \ifx\@empty\eql@scan@stack
```

```

3463 \checkend{#2}%
3464 \eql@scan@addto{#1}%
3465 \expandafter\let\eql@scan@iterate\eql@scan@env@org
3466 (dev)\eql@dev@leave\eql@scan@env
3467 \expandafter\eql@scan@end
3468 \else
3469 \eql@scan@addto{#1\end{#2}}%
3470 \expandafter\eql@scan@iterate
3471 \fi
3472 }

```

`\an@env@iterate@nopar` Version of `\eql@scan@env@iterate` which does not accept `\par` within the argument:

```

3473 \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.

```

3474 \long\def\eql@scan@env@count#1\begin#2{%
3475 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
3476 }

```

The call-back macro `\eql@scan@env@cancel` ignores the body as well as the end clause for the environment:

```

3477 \def\eql@scan@env@cancel{%
3478 \@namedef{end\@currentvir}{\ignorespacesafterend}%
3479 }

```

**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 `\[`:

```

3480 \def\eql@scan@sqr#1{%
3481 (dev)\eql@dev@enter\eql@scan@sqr
3482 \def\eql@scan@end{#1\]}%
3483 \eql@scan@reg@{\def\eql@scan@stack{b}%
3484 \let\eql@scan@sqr@org\[%\]}
3485 \ifdefined\eql@scan@par
3486 \let\[\eql@scan@sqr@iterate%\]
3487 \else
3488 \let\[\eql@scan@sqr@iterate@nopar%\]
3489 \fi
3490 \[%\]
3491 }

```

`\eql@scan@sqr@iterate` Iterate until we find a balanced pairing of square brackets. Then call the call-back macro:

```

3492 \long\def\eql@scan@sqr@iterate#1\]{%
3493 \edef\eql@scan@stack{%
3494 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
3495 \ifx\@empty\eql@scan@stack
3496 \let\[\eql@scan@sqr@org%\]
3497 \eql@scan@addto{#1}%
3498 (dev)\eql@dev@leave\eql@scan@sqr
3499 \expandafter\eql@scan@end

```

```

3500 \else
3501   \eql@scan@addto{#1\}}%
3502   \expandafter\[%\]
3503 \fi
3504 }

```

`\an@sqr@iterate@nopar` Version of `\eql@scan@sqr@iterate` which does not accept `\par` within the argument:

```

3505 \def\eql@scan@sqr@iterate@nopar#1\{\eql@scan@sqr@iterate#1\}}

```

`\eql@scan@sqr@count` Push a ‘b’ for every encountered instance of ‘\[':

```

3506 \long\def\eql@scan@sqr@count#1\[#2{\% \]
3507   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
3508 }

```

`\l@scan@sqrang@cancel` The call-back macro `\eql@scan@sqrang@cancel` ignores the body and the closing bracket:

```

3509 \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 `\>`:

```

3510 \def\eql@scan@ang#1{%
3511 (dev)\eql@dev@enter\eql@scan@ang
3512   \def\eql@scan@end{#1\>}%
3513   \eql@scan@reg@{\}\def\eql@scan@stack{b}%
3514   \let\eql@scan@ang@org\<%\>
3515   \ifdefined\eql@scan@par
3516     \let\<\eql@scan@ang@iterate%\>
3517   \else
3518     \let\<\eql@scan@ang@iterate@nopar%\>
3519   \fi
3520   \<%\>
3521 }

```

`\eql@scan@ang@iterate` Iterate until we find a balanced pairing of angle brackets:

```

3522 \long\def\eql@scan@ang@iterate#1\>{%
3523   \edef\eql@scan@stack{%
3524     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
3525   \ifx\@empty\eql@scan@stack
3526     \let\<\eql@scan@ang@org%\>
3527     \eql@scan@addto{#1}%
3528 (dev)\eql@dev@leave\eql@scan@ang
3529   \expandafter\eql@scan@end
3530   \else
3531     \eql@scan@addto{#1\>}%
3532     \expandafter\<%\>
3533   \fi
3534 }

```

`\an@ang@iterate@nopar` Version of `\eql@scan@ang@iterate` which does not accept `\par` within the argument:

```

3535 \def\eql@scan@ang@iterate@nopar#1\>\eql@scan@ang@iterate#1\>}

```

`\eql@scan@ang@count` Push a ‘b’ for every encountered instance of ‘<’:

```
3536 \long\def\eql@scan@ang@count#1<#2{%\>
3537 \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
3538 }
```

## O.2 Options Processing

`\eql@equations@testall` The macro sequence started by `\eql@equations@testall` scans for optional arguments to the equation environments and appends them to the argument list using `\eqnaddopt`. The argument scheme is roughly `{ !t~ !t* !t! !o !e{@} }`. All arguments are scanned such that any spaces stop the scanning and such that any alignment markers ‘&’ cannot interfere:

```
3539 \def\eql@equations@testall{\eql@equations@testtilde}
3540 \def\eql@equations@testtilde#1{%
3541 \eql@ifnextgobble@tight~%
3542 {\eqnaddopt{lines}\eql@equations@testopt{#1}}%
3543 {\eql@equations@testopt{#1}}}
3544 \def\eql@equations@testopt#1{%
3545 \eql@ifnextchar@tight[%
3546 {\eql@equations@addopt{\eql@equations@testexcl{#1}}}%
3547 {\eql@equations@testexcl{#1}}}
3548 \def\eql@equations@addopt#1[#2]{\eqnaddopt{#2}#1}
3549 \def\eql@equations@testexcl#1{%
3550 \eql@ifnextgobble@tight!%
3551 {\eqnaddopt{donumber}\eql@equations@testat{#1}}}%
3552 {\eql@equations@teststar{#1}}}
3553 \def\eql@equations@teststar#1{%
3554 \eql@ifstar@tight%
3555 {\eqnaddopt{nonumber}\eql@equations@testat{#1}}}%
3556 {\eql@equations@testat{#1}}}
3557 \def\eql@equations@testat#1{%
3558 \eql@ifat@tight
3559 {\eql@equations@addlabel{#1}}}%
3560 {#1}}
3561 \def\eql@equations@addlabel#1#2{\eqnaddopt{label={#2}}#1}
```

`\eql@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:

```
3562 \def\eql@equations@processopt{%
3563 \let\eql@blocklabel\@undefined
3564 \let\eql@blocklabelname\@undefined
3565 \let\eql@blocktag\@undefined
3566 \let\eql@skip@force@above\@undefined
3567 \let\eql@skip@force@below\@undefined
3568 \let\eql@skip@force@leave\@undefined
3569 \let\eql@display@linewidth\@undefined
3570 \let\eql@display@marginleft\@undefined
3571 \let\eql@display@marginright\@undefined
3572 \eql@abovespace@z@skip
3573 \eql@belowspace@z@skip
3574 \eql@displaybreak@prepen@\@MM
3575 \eql@displaybreak@postpen@\@MM
3576 \eql@frame@reset}
```

```

3577 \eql@nextopt@process{equations}%
3578 \let\eql@punct@block\eql@punct@main
3579 \let\eql@punct@main\relax
3580 \eql@indent@glueexpr\eql@indent@val\relax
3581 \eql@tagsepmin@glueexpr\eql@tagsepmin@val\relax
3582 }

```

### 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:

```

3583 \def\eql@single@cr{%
3584 \eql@error{Cannot use '\string\\' within display equation.
3585 Please switch to equations environment}%
3586 }

```

`\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:

```

3587 \def\eql@single@start{%
3588 \eql@display@enter
3589 \eql@tagging@start
3590 \eql@dollar@begin
3591 \eql@display@adjust
3592 \eql@numbering@mode@eval
3593 \eql@stack@save@single
3594 \eql@numbering@single@init
3595 \ifdefined\eql@single@crerror\else
3596 \let\\ \eql@single@cr
3597 \fi
3598 \ifdefined\eql@single@native
3599 \let\eql@single@start@sel\eql@single@start@native
3600 \let\eql@single@end@sel\eql@single@end@native
3601 \else
3602 \let\eql@single@start@sel\eql@single@start@print
3603 \let\eql@single@end@sel\eql@single@end@print
3604 \fi
3605 \eql@single@start@sel
3606 }

```

`\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:

```

3607 \def\eql@single@end{%
3608 \eql@punct@apply@block
3609 \eql@hook@eqout
3610 \eql@single@end@sel
3611 \eql@stack@restore
3612 \eql@dollar@end
3613 \eql@tagging@end
3614 \eql@display@leave
3615 }

```



`\eql@single@main` Combined opening, body and closing for pre-scanned body: **TODO:** is `\expandafter` needed? relic?

```
3616 \def\eql@single@main{%
3617   \expandafter\eql@single@start
3618   \eql@scan@body
3619   \eql@single@end
3620 }
```

`\eql@mode@single` Configure equations macros to single-line mode:

```
3621 \def\eql@mode@single{%
3622   \ifdefined\eql@single@doscan
3623     \let\eql@equations@main\eql@single@main
3624     \let\eql@equations@end\@empty
3625   \else
3626     \let\eql@equations@main\@undefined
3627     \let\eql@equations@end\eql@single@end
3628   \fi
3629 }
```

## O.4 Multi-Line Main

`\multi@mode@lines` (*bool*) Switch register for lines vs. columns mode:

```
3630 \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

```
3631 \def\eql@multi@main{%
3632   \eql@display@enter
3633   \eql@tagging@start
3634   \eql@dollar@dollar@begin
3635   \eql@display@adjust
3636   \eql@numbering@mode@eval
3637   \eql@stack@save@multi
3638   \ifdefined\eql@numbering@subeq@use
3639     \eql@numbering@subeq@init
3640   \fi
3641   \eql@display@init
3642   \let\intertext\eql@intertext
3643   \let\endintertext\endeql@intertext
3644   \eql@shape@align@enable
```

Now measure the given multi-line equations body:

```
3645   \ifdefined\eql@multi@mode@lines
3646     \eql@lines@measure
3647   \else
3648     \ifdefined\eql@ampproof@active
3649       \eql@ampproof
3650     \fi
3651     \eql@columns@measure
3652   \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`:

```

3653 \ifx\eql@numbering@subeq@use\@ne
3654   \eql@numbering@subeq@revert
3655 \fi
3656 \ifdefined\eql@multi@mode@lines\else
3657   \ifdefined\eql@multi@linesfallback
3658     \ifnum\eql@totalcolumns@=\@ne
3659       \let\eql@multi@mode@lines\eql@true
3660       \ifx\eql@multi@linesfallback\z@\else
3661         \eql@lines@measure
3662       \fi
3663     \fi
3664   \fi
3665 \fi

```

Adjust the multi-line equations body:

```

3666 \ifdefined\eql@multi@mode@lines
3667   \eql@lines@adjust
3668 \else
3669   \eql@columns@adjust
3670 \fi

```

Now print the multi-line equations body:

```

3671 \eql@display@print
3672 \eql@numbering@print@init
3673 \ifdefined\eql@multi@mode@lines
3674   \eql@lines@print
3675 \else
3676   \eql@columns@print
3677 \fi
3678 \eql@display@close

```

Close numbering, restore global variables, end display math, indicate end to pdf tagging, return to vertical mode if desired:

```

3679 \ifdefined\eql@numbering@subeq@use
3680   \eql@numbering@subeq@close
3681 \fi
3682 \eql@stack@restore
3683 \eql@dollar@dollar@end
3684 \eql@tagging@end
3685 \eql@display@leave
3686 }

```

`\eql@mode@columns` Configure equations macros to one of the two multi-line modes:

```

\eql@mode@lines
3687 \def\eql@mode@columns{%
3688   \let\eql@equations@main\eql@multi@main
3689   \let\eql@equations@end\@empty
3690   \let\eql@multi@mode@lines\eql@false
3691 }
3692 \def\eql@mode@lines{%
3693   \let\eql@equations@main\eql@multi@main
3694   \let\eql@equations@end\@empty
3695   \let\eql@multi@mode@lines\eql@true
3696 }

```

## 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`:

```

3697 \newenvironment{equations}{%
3698 (dev)\eql@dev@enterenv
3699   \ifmmode
3700     \eqLError@mathmode{\string\begin{\@currenvir}}%
3701     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
3702   \else
3703     \expandafter\eql@ampprotect\expandafter\eqlequations@testall
3704     \expandafter\eqlequations@start
3705   \fi
3706 }{%
3707   \eqlequations@end
3708   \ignorespacesafterend
3709 (dev)\eql@dev@leaveenv
3710 }
```

`\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`:

```

3711 \def\eqlequations@start{%
3712   \eqlequations@processopt
3713   \ifdefined\eqlequations@main
3714     \expandafter\eql@scan@env\expandafter\eqlequations@main
3715   \else
3716     \expandafter\eql@single@start
3717   \fi
3718 }
```

### Square Brackets.

**equations@sqr** (*env.*) Define a pseudo-environment `equations@sqr` such that `\@currenvir` may point to it when needed:

```

3719 \newenvironment{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`:

```

3720 \protected\def\eqlequations@sqr@open{%
3721   \ifmmode
3722     \eqLError@mathmode{\string\[...\string\]}%
3723     \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
3724   \else
3725 (dev)\eql@dev@enter{\[...\string\]}%
3726     \expandafter\eqnaddopt\expandafter{\eqlequations@sqr@opt}%
3727     \begin{equations@sqr}%

```

```

3728 \let\}\eqlequations@sqr@close
3729 \expandafter\eql@ampprotect\expandafter\eqlequations@testall
3730 \expandafter\eqlequations@sqr@start
3731 \fi
3732 }

```

`@equations@sqr@start` Process arguments. Depending on mode of operation, scan and process enclosed contents via `\eqlequations@main` or pass on to `\eql@single@start`:

```

3733 \def\eqlequations@sqr@start{%
3734 \eqlequations@processopt
3735 \ifdefined\eqlequations@main
3736 \expandafter\eql@scan@sqr\expandafter\eqlequations@main
3737 \else
3738 \expandafter\eql@single@start
3739 \fi
3740 }

```

`@equations@sqr@close` Definition for ‘\’:

```

3741 \protected\def\eqlequations@sqr@close{%
3742 \eqlequations@end
3743 (dev)\eql@dev@leave{\[...\string\]}%
3744 \end{equations@sqr}%
3745 \ignorespaces
3746 }

```

**TODO:** describe

```

\eql@sqr@open
\eql@sqr@close
3747 \let\eql@sqr@open\eqlequations@sqr@open
3748 \protected\def\eql@sqr@close{%
3749 \eql@error{'\string\'} may only close '\string\[\'%]
3750 }

```

## Angle Brackets.

`equations@ang` (*env.*) Define a pseudo-environment `equations@ang`:

```

3751 \newenvironment{equations@ang}{}{}
3752 \newenvironment{equations@ang}{}{}

```

`\eql@ang@open` Definition for ‘<’. Forward to `equationsbox` if in math mode, otherwise to `equations`:

```

3753 \protected\def\eql@ang@open{%
3754 (dev)\eql@dev@enter{\<...\string\>}%
3755 \ifmmode
3756 \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
3757 \begin{equationsbox@ang}%
3758 \let\>\eql@box@ang@close
3759 \expandafter\eql@ampprotect\expandafter\eql@box@testall
3760 \expandafter\eql@box@start
3761 \else
3762 \expandafter\eqnaddopt\expandafter{\eqlequations@ang@opt}%
3763 \begin{equations@ang}%
3764 \let\>\eqlequations@ang@close
3765 \expandafter\eql@ampprotect\expandafter\eqlequations@testall
3766 \expandafter\eqlequations@ang@start

```

```

3767 \fi
3768 }

```

`\eql@ang@close` Definition for ‘>’: **TODO:** NOTE: `\protected` acts as `\relax` and starts a row in `\halign`, so we overwrite `\>` when starting.

```

3769 \protected\def\eql@ang@close{%
3770 \eql@error{'\string\>' may only close '\string\<'}%
3771 }

```

`@equations@ang@start` Process arguments and start handling the equation:

```

3772 \def\eql@equations@ang@start{%
3773 \eql@equations@processopt
3774 \ifdefined\eql@equations@main
3775 \expandafter\eql@scan@ang\expandafter\eql@equations@main
3776 \else
3777 \expandafter\eql@single@start
3778 \fi
3779 }

```

`@equations@ang@close` **TODO:** describe

```

3780 \def\eql@equations@ang@close{%
3781 \eql@equations@end
3782 \end{equations@ang}%
3783 (dev)\eql@dev@leave{\<...\string\>}%
3784 \ignorespaces
3785 }

```

`\eql@box@ang@close` **TODO:** describe

```

3786 \def\eql@box@ang@close{%
3787 \eql@box@end
3788 \end{equationsbox@ang}%
3789 (dev)\eql@dev@leave{\<...\string\>}%
3790 \ignorespaces
3791 }

```

## P Options

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

```

3792 \RequirePackage{keyval}

```

### P.1 Selection Tools

`\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:

```

3793 \def\eq1@decide@relax{\@tempb:=\relax}
3794 \def\eq1@decide@select#1#2#3#4{%
3795   \def\@tempa{#3}%
3796   \let\@tempd\@undefined
3797   \@for\@tempc:=#4\do{%
3798     \ifdefined\@tempd\else
3799       \edef\@tempb{\noexpand\@tempb:=\expandafter\@firstoftwo\@tempc}%
3800       \ifx\@tempb\eq1@decide@relax
3801         \def\@tempa{\relax}%
3802       \fi
3803       \expandafter\@for\@tempb\do{%
3804         \ifx\@tempa\@tempb
3805           \expandafter\expandafter\expandafter\def
3806             \expandafter\expandafter\expandafter\@tempd
3807               \expandafter\expandafter\expandafter{%
3808                 \expandafter\@secondoftwo\@tempc}%
3809         \fi
3810       }%
3811     \fi
3812   }%
3813   \ifdefined\@tempd
3814     \@tempd
3815   \else
3816     \eq1@error{undefined value ‘#3’ for option ‘#2’ of ‘#1’}%
3817   \fi
3818 }

```

`\eq1@decide@if` Decide between true and false or related pairs of values:

```

3819 \def\eq1@decide@true{on,true,yes,enabled,1}
3820 \def\eq1@decide@false{off,false,no,disabled,0}
3821 \def\eq1@decide@if#1#2#3#4#5{%
3822   \eq1@decide@select{#1}{#2}{#3}{%
3823     {\eq1@decide@true{#4}},%
3824     {\eq1@decide@false{#5}}}%

```

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

```

3825 \def\eq1@decide@bool#1#2#3#4{%
3826   \eq1@decide@if{#1}{#2}{#3}{\let#4\eq1@true}{\let#4\eq1@false}}

```

`\eq1@decide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```

3827 \def\eq1@decide@abovebelow#1#2#3#4#5{%
3828   \eq1@decide@select{#1}{#2}{#3}{%
3829     {,abovebelow,both,tb}{#4#5},%
3830     {above,top,t}{#4},%
3831     {below,bottom,b}{#5}}%

```

`\eq1@decide@situation` Select a particular vertical spacing situation and store it in the macro #4:

```

3832 \def\eq1@decide@situation#1#2#3#4{%
3833   \eq1@decide@select{#1}{#2}{#3}{%
3834     {{long}}{\def#4{0}}},%

```

```

3835    {{short}}{\def#4{1}}},%
3836    {{cont}}{\def#4{2}}},%
3837    {{par}}{\def#4{3}}},%
3838    {{top}}{\def#4{4}}},%
3839    {{noskip}}{\def#4{5}}},%
3840    {{medskip}}{\def#4{6}}}}

```

## P.2 Declaration Code

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

```

3841 \def\eql@define@key#1#2{%
3842   \eql@ifnextchar@loose[%]
3843     {\eql@definekey@opt{#1}{#2}}%
3844     {\eql@definekey@noopt{#1}{#2}}%
3845 }
3846 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{{#3}}}
3847 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{{#3}{#4}}}
3848 \def\eql@definekey@for#1#2#3{%
3849   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}{#3}%
3850     \edef\eql@for@var{ \noexpand\eql@for@var:=#1}%
3851     \expandafter\@for\eql@for@var\do{%
3852       \edef\eql@for@varb{\noexpand\eql@for@var:=#2}%
3853       \expandafter\@for\eql@for@varb\do{%
3854         \edef\eql@for@call##1{%
3855           \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@var}}%
3856         \eql@for@call{##1}%
3857       }%
3858     }%
3859 }

```

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

```

3860 \def\eql@setkeys#1#2{%
3861   \def\eql@tmp{\setkeys{eql@#1}}%
3862   \expandafter\eql@tmp\expandafter{#2}%
3863 }

```

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

```

\eqnaddopt
3864 \let\eql@nextopt\@empty
3865 \def\eql@nextopt@process#1{%
3866   (dev)\eql@dev@start\eql@nextopt@process
3867   \eql@setkeys{#1}\eql@nextopt
3868   \let\eql@tagging@opt\eql@nextopt
3869   \global\let\eql@nextopt\@empty
3870 }
3871 \newcommand{\eqnaddopt}[1]{%
3872   \expandafter\def\expandafter\eql@nextopt\expandafter{\eql@nextopt,#1}}

```

## P.3 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:

```

3873 \eqld@define@key{equationsbox}{gathered,gather,ga,lines,ln,\string~}[]{%
3874   \eqld@mode@stacked}
3875 \eqld@define@key{equationsbox}{aligned,align,al,columns,col,@}[]{%
3876   \eqld@mode@aligned}
3877 \eqld@define@key{equationsbox}{top,t}[]{\let\eqld@box@box\vtop}
3878 \eqld@define@key{equationsbox}{center,c}[]{\let\eqld@box@box\vcenter}
3879 \eqld@define@key{equationsbox}{bottom,b}[]{\let\eqld@box@box\vbox}
3880 \eqld@define@key{equationsbox}{frame}{\fbox}{%
3881   \def\eqld@box@frame{#1}}%
3882   \ifx\eqld@box@frame\@empty\let\eqld@box@frame\@firstofone\fi}
3883 \eqld@define@key{equationsbox}{wrap}[]{\eqld@box@wrap#1}
3884 \eqld@define@key{setup}{boxangopt}[]{%
3885   \def\eqld@box@ang@opt{columns,#1}}

```

**Modes for Equations Environment.** Declare modes and switches for the equations environment:

```

3886 \eqld@define@key{equations}{equation,eq,single,1}[]{\eqld@mode@single}
3887 \eqld@define@key{equations}{gathered,gather,ga,lines,ln,\string~}[]{%
3888   \eqld@mode@lines}
3889 \eqld@define@key{equations}{aligned,align,al,columns,col,@}[]{%
3890   \eqld@mode@columns}
3891 \eqld@define@key{equations}{native}[true]{%
3892   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native%
3893   \ifdefined\eqld@single@native\let\eqld@layoutleft\eqld@false\fi}
3894 \eqld@define@key{setup}{native}[true]{%
3895   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native}
3896 \eqld@define@key{setup}{scanequation}[true]{%
3897   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@doscan}
3898 \eqld@define@key{setup}{sqropt}[]{%
3899   \def\eqld@equations@sqr@opt{equation,#1}}
3900 \eqld@define@key{setup}{angopt}[]{%
3901   \def\eqld@equations@ang@opt{columns,#1}}

```

**Vertical Spacing.** Settings concerning the spacing of lines: **TODO:** set at end of env only!

```

3902 \def\eqld@keycat{equations,equationsbox,setup}
3903 \eqld@define@key\eqld@keycat{spread}{\def\eqld@spread@val{#1}}
3904 \eqld@define@key\eqld@keycat{strut}[true]{\eqld@decide@select{#3}{#2}{#1}{%
3905   {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%
3906   {{cell}{\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\relax}},%
3907   {{tag}{\let\eqld@strut@cell\relax\let\eqld@strut@tag\eqld@strut}},%
3908   {\eqld@decide@true
3909     {\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\eqld@strut}}}}
3910 \eqld@define@key{setup}{strutdepth}{\def\eqld@strut@depth{#1}}

```

Settings concerning page breaks:

```

3911 \eqld@define@key{equations}{prebreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
3912   {\eqld@decide@true{\eqld@displaybreak@pre4}},%
3913   {{force}{\eqld@displaybreak@pre4}},%
3914   {{high}{\eqld@displaybreak@pre3}},%
3915   {{med,medium}{\eqld@displaybreak@pre2}},%
3916   {{low}{\eqld@displaybreak@pre1}},%

```



```

3917 \eqld@decide@false{\eqld@displaybreak@pre0}},%
3918 {{default,inherit,0}{\eqld@displaybreak@pre\m@ne}},%
3919 {\relax{\eqld@displaybreak@pre{#1}}}}
3920 \eqld@define@key{equations}{postbreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
3921 \eqld@decide@true{\eqld@displaybreak@post4}},%
3922 {{force}{\eqld@displaybreak@post4}},%
3923 {{high}{\eqld@displaybreak@post3}},%
3924 {{med,medium}{\eqld@displaybreak@post2}},%
3925 {{low}{\eqld@displaybreak@post1}},%
3926 {\eqld@decide@false{\eqld@displaybreak@post0}},%
3927 {{default,inherit,}{\eqld@displaybreak@post\m@ne}},%
3928 {\relax{\eqld@displaybreak@post{#1}}}}
3929 \eqld@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
3930 \eqld@decide@select{#3}{#2}{#1}{%
3931 {{full}{\eqld@displaybreak@inter4}},%
3932 {{high}{\eqld@displaybreak@inter3}},%
3933 {{med,medium}{\eqld@displaybreak@inter2}},%
3934 {{low}{\eqld@displaybreak@inter1}},%
3935 {\eqld@decide@false{\eqld@displaybreak@inter\z@}},%
3936 {\relax{\eqld@displaybreak@inter{#1}}}}
3937 \eqld@define@key{equations}{prepenalty}{%
3938 \eqld@displaybreak@prepen\numexpr#1\relax}
3939 \eqld@define@key{equations}{postpenalty}{%
3940 \eqld@displaybreak@postpen\numexpr#1\relax}
3941 \eqld@define@key{equations,setup}{interpenalty}{%
3942 \interdisplaylinepenalty\numexpr#1\relax}

```

Settings to specify the apparent height and depth of equations:

```

3943 \eqld@define@key\eqld@keycat{displayheight}[strut]{%
3944 \eqld@decide@select{#3}{#2}{#1}{%
3945 {\eqld@decide@false{\let\eqld@display@height\@undefined}},%
3946 {{strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
3947 {\relax{\def\eqld@display@height{#1}}}}
3948 \eqld@define@key\eqld@keycat{displaydepth}[strut]{%
3949 \eqld@decide@select{#3}{#2}{#1}{%
3950 {\eqld@decide@false{\let\eqld@display@depth\@undefined}},%
3951 {{strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
3952 {\relax{\def\eqld@display@depth{#1}}}}

```

Override vertical spacing situation: **TODO**: short should just apply to above?! or as far as short would apply...

```

3953 \eqld@define@key{equations}{noskip}[]{%
3954 \eqld@decide@abovebelow{#3}{#2}{#1}%
3955 {\def\eqld@skip@force@above{5}}%
3956 {\def\eqld@skip@force@below{5}}
3957 \eqld@define@key{equations}{short}[above]{%
3958 \eqld@decide@abovebelow{#3}{#2}{#1}%
3959 {\def\eqld@skip@force@above{1}}%
3960 {\def\eqld@skip@force@below{1}}
3961 \eqld@define@key{equations}{long}[]{%
3962 \eqld@decide@abovebelow{#3}{#2}{#1}%
3963 {\def\eqld@skip@force@above{0}}%
3964 {\def\eqld@skip@force@below{0}}
3965 \eqld@define@key{equations}{medskip}[]{%
3966 \eqld@decide@abovebelow{#3}{#2}{#1}%
3967 {\def\eqld@skip@force@above{6}}%
3968 {\def\eqld@skip@force@below{6}}
3969 \eqld@define@key{equations}{par}[par]{%

```

```

3970 \eql@decide@select{#3}{#2}{#1}{%
3971   {{default,}\let\eql@skip@force@leave\@undefined}},%
3972   {{cont,hmode}\let\eql@skip@force@leave\z@}},%
3973   {{par,vmode}\let\eql@skip@force@leave\@ne
3974     \ifdefined\eql@skip@force@below\else
3975       \def\eql@skip@force@below{3}%
3976     \fi}},%
3977   {{top}\let\eql@skip@force@leave\tw@
3978     \ifdefined\eql@skip@force@below\else
3979       \def\eql@skip@force@below{4}
3980     \fi}}}}

```

Specify vertical spacing explicitly:

```

3981 \eql@define@key{equations}{skip}{%
3982   \def\eql@skip@force@above{7}%
3983   \def\eql@skip@custom@above{#1}%
3984   \let\eql@skip@force@below\eql@skip@force@above
3985   \let\eql@skip@custom@below\eql@skip@custom@above}
3986 \eql@define@key{equations}{aboveskip}{%
3987   \def\eql@skip@force@above{7}%
3988   \def\eql@skip@custom@above{#1}}
3989 \eql@define@key{equations}{belowskip}{%
3990   \def\eql@skip@force@below{7}%
3991   \def\eql@skip@custom@below{#1}}
3992 \eql@define@key{equations}{abovespace}{%
3993   \advance\eql@abovespace@glueexpr#1\relax}
3994 \eql@define@key{equations}{belowspace}{%
3995   \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

3996 \eql@define@key{intertext}{skip}{%
3997   \def\eql@skip@force@above{7}%
3998   \def\eql@skip@custom@above{#1}%
3999   \let\eql@skip@force@below\eql@skip@force@above
4000   \let\eql@skip@custom@below\eql@skip@custom@above}
4001 \eql@define@key{intertext}{aboveskip}{%
4002   \def\eql@skip@force@below{7}%
4003   \def\eql@skip@custom@below{#1}}
4004 \eql@define@key{intertext}{belowskip}{%
4005   \def\eql@skip@force@above{7}%
4006   \def\eql@skip@custom@above{#1}}
4007 \eql@define@key{intertext}{noskip}[]{%
4008   \eql@decide@abovebelow{#3}{#2}{#1}%
4009   {\def\eql@skip@force@below{5}}%
4010   {\def\eql@skip@force@above{5}}}
4011 \eql@define@key{intertext}{short}[]{%
4012   \eql@decide@abovebelow{#3}{#2}{#1}%
4013   {\def\eql@skip@force@below{1}}%
4014   {\def\eql@skip@force@above{1}}}
4015 \eql@define@key{intertext}{long}[]{%
4016   \eql@decide@abovebelow{#3}{#2}{#1}%
4017   {\def\eql@skip@force@below{0}}%
4018   {\def\eql@skip@force@above{0}}}
4019 \eql@define@key{intertext}{medskip}[]{%
4020   \eql@decide@abovebelow{#3}{#2}{#1}%
4021   {\def\eql@skip@force@below{6}}%
4022   {\def\eql@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

4023 \eqld@define@key{setup}{skip,longskip}{%
4024   \abovedisplayskip\glueexpr#1\relax
4025   \belowdisplayskip\abovedisplayskip
4026   \def\eqld@skip@long@above{#1}%
4027   \let\eqld@skip@long@below\eqld@skip@long@above}
4028 \eqld@define@key{setup}{aboveskip,abovelongskip}{%
4029   \abovedisplayskip\glueexpr#1\relax
4030   \def\eqld@skip@long@above{#1}}
4031 \eqld@define@key{setup}{belowskip,belowlongskip}{%
4032   \belowdisplayskip\glueexpr#1\relax
4033   \def\eqld@skip@long@below{#1}}
4034 \eqld@define@key{setup}{aboveshortskip}{%
4035   \abovedisplayshortskip\glueexpr#1\relax
4036   \def\eqld@skip@short@above{#1}}
4037 \eqld@define@key{setup}{belowshortskip}{%
4038   \belowdisplayshortskip\glueexpr#1\relax
4039   \def\eqld@skip@short@below{#1}}
4040 \eqld@define@key{setup}{tagskip}{%
4041   \def\eqld@skip@tag@above{#1}%
4042   \let\eqld@skip@tag@below\eqld@skip@tag@above}
4043 \eqld@define@key{setup}{abovetagskip}{%
4044   \def\eqld@skip@tag@above{#1}}
4045 \eqld@define@key{setup}{belowtagskip}{%
4046   \def\eqld@skip@tag@below{#1}}
4047 \eqld@define@key{setup}{medskip}{%
4048   \def\eqld@skip@med@above{#1}%
4049   \let\eqld@skip@med@below\eqld@skip@med@above}
4050 \eqld@define@key{setup}{abovemedskip}{%
4051   \def\eqld@skip@med@above{#1}}
4052 \eqld@define@key{setup}{belowmedskip}{%
4053   \def\eqld@skip@med@below{#1}}
4054 \eqld@define@key{setup}{medtagskip}{%
4055   \def\eqld@skip@medtag@above{#1}%
4056   \let\eqld@skip@medtag@below\eqld@skip@medtag@above}
4057 \eqld@define@key{setup}{abovemedtagskip}{%
4058   \def\eqld@skip@medtag@above{#1}}
4059 \eqld@define@key{setup}{belowmedtagskip}{%
4060   \def\eqld@skip@medtag@below{#1}}
4061 \eqld@define@key{setup}{abovetopskip}{%
4062   \def\eqld@skip@top@above{#1}}
4063 \eqld@define@key{setup}{belowtopskip}{%
4064   \def\eqld@skip@top@below{#1}}
4065 \eqld@define@key{setup}{aboveparskip}{%
4066   \def\eqld@skip@par@above{#1}}
4067 \eqld@define@key{setup}{belowparskip}{%
4068   \def\eqld@skip@par@below{#1}}
4069 \eqld@define@key{setup}{abovepartagskip}{%
4070   \def\eqld@skip@partag@above{#1}}
4071 \eqld@define@key{setup}{belowpartagskip}{%
4072   \def\eqld@skip@partag@below{#1}}
4073 \eqld@define@key{setup}{abovecontskip}{%
4074   \eqld@decide@select{#3}{#2}{#1}{%
4075     {\hide}{\def\eqld@skip@cont@above{\eqld@spread@val-\eqld@skip@long@below}}},%
4076     {\relax{\def\eqld@skip@cont@above{#1}}}}}%
4077 \eqld@define@key{setup}{belowcontskip}{%
4078   \def\eqld@skip@cont@below{#1}}
4079 \eqld@define@key{setup}{shortmode}{%

```

```

4080 \eql@decide@select{#3}{#2}{#1}{%
4081   {{off,never,no}{\def\eql@skip@mode@short{0}}},%
4082   {{above,neverbelow,notbelow,belowoff}{\def\eql@skip@mode@short{1}}},%
4083   {{belowone,belowsingle}{\def\eql@skip@mode@short{2}}},%
4084   {{belowall,always,on}{\def\eql@skip@mode@short{3}}}}
4085 \eql@define@key{setup}{abovecontmode}{%
4086   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}
4087 \eql@define@key{setup}{belowcontmode}{%
4088   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
4089 \eql@define@key{setup}{aboveparmode}{%
4090   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
4091 \eql@define@key{setup}{belowparmode}{%
4092   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
4093 \eql@define@key{setup}{abovetopmode}{%
4094   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
4095 \eql@define@key{setup}{belowtopmode}{%
4096   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

**Labels and Tag Declaration.** Specify label and tag for equations and subequations:

```

4097 \def\eql@keycat{equations,subequations}
4098 \eql@define@key\eql@keycat{label}{\eql@blocklabel@set{#1}}
4099 \eql@define@key\eql@keycat{labelname}{\eql@blocklabelname@set{#1}}
4100 \eql@define@key\eql@keycat{tag}{\eql@blocktag@set\one{#1}}
4101 \eql@define@key\eql@keycat{tag*}{\eql@blocktag@set\z@{#1}}
4102 \eql@define@key{setup}{labelname}{\protected@edef\eql@labelname@generic{#1}}
4103 \eql@define@key{setup}{autolabel}[true]{%
4104   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@autolabel}
4105 \eql@define@key{setup}{autotag}[true]{%
4106   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@autotag}

```

**Tag Spacing.** Configure horizontal spacing for equation tags:

```

4107 \def\eql@keycat{equations,setup}
4108 \eql@define@key\eql@keycat{tagmargin}[auto]{%
4109   \eql@decide@select{#3}{#2}{#1}{%
4110     {\eql@decide@false{\let\eql@tagmargin@val\@undefined}},%
4111     {{auto}{\let\eql@tagmargin@val\@undefined}},%
4112     {\relax{\def\eql@tagmargin@val{#1}}}}}
4113 \eql@define@key\eql@keycat{tagmargin*}{%
4114   \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}}
4115 \eql@define@key\eql@keycat{tagmarginratio}{%
4116   \eql@tagmargin@ratio@\dimexpr#1pt\relax}
4117 \eql@define@key\eql@keycat{tagmarginthreshold}{%
4118   \def\eql@tagmargin@threshold{#1}}
4119 \eql@define@key\eql@keycat{mintagsep}{\def\eql@tagsepmin@val{#1}}
4120 \eql@define@key\eql@keycat{mintagwidth}{%
4121   \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}}
4122 \eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin{#1}}

```

**Tag Layout.** Configure methods to declare equation tag layout:

```

4123 \eql@define@key{setup}{tagbox,taglayout}{\eql@tag@setbox{#1}}
4124 \eql@define@key{setup}{tagbox*,taglayout*}{\eql@tag@setbox@{#1}}
4125 \eql@define@key{setup}{tagform}{\eql@tag@setform#1}
4126 \eql@define@key{setup}{tagform*}{\eql@tag@setform@{#1}}
4127 \eql@define@key{setup}{subeqtemplate}{%

```

```

4128 \def\eql@subequations@template####1####2{#1}%
4129 \expandafter\def\expandafter\eql@subequations@template\expandafter{%
4130   \eql@subequations@template\theparentequation{equation}}}%
4131 }

```

**Equation Numbering.** Configure equation numbering schemes:

```

4132 \def\eql@keycat{equations,setup}
4133 \eql@define@key\eql@keycat{numberline,numline,n}[all]{%
4134   \eql@numbering@set{##1}}
4135 \eql@define@key\eql@keycat{nonumber,nn,*}[]{%
4136   \let\eql@numbering@active\eql@false}
4137 \eql@define@key\eql@keycat{donumber,dn,!}[]{%
4138   \let\eql@numbering@active\eql@true}
4139 \eql@define@key\eql@keycat{number,num}[true]{%
4140   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@active}
4141 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
4142 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
4143 \eql@define@key\eql@keycat{tags,eqno}{%
4144   \eql@decide@select{#3}{#2}{#1}{%
4145     {\right,r}{\let\eql@tagsleft\eql@false}},%
4146     {\left,l}{\let\eql@tagsleft\eql@true}}}}
4147 \eql@define@key\eql@keycat{bestlineauto}[true]{%
4148   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}

```

**Horizontal Layout.** Configure horizontal alignment mode and margin for left alignment:

```

4149 \def\eql@keycat{equations,setup}
4150 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
4151   {{center,c}{\let\eql@layoutleft\eql@false}},%
4152   {{left,l}{\let\eql@layoutleft\eql@true}}}}
4153 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
4154 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
4155 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin{#1}}
4156 \eql@define@key\eql@keycat{leftmargin*}{%
4157   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin{\the\dimen@}}
4158 \eql@define@key\eql@keycat{minleftmargin}{%
4159   \def\eql@layoutleftmarginmin{#1}}
4160 \eql@define@key\eql@keycat{maxleftmargin}{%
4161   \eql@decide@select{#3}{#2}{#1}{%
4162     {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
4163     {\relax{\def\eql@layoutleftmarginmax{#1}}}}}
4164 \def\eql@keycat{equations,equationsbox}
4165 \eql@define@key\eql@keycat{margin}{%
4166   \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
4167 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}
4168 \eql@define@key\eql@keycat{marginright}{\def\eql@display@marginright{#1}}
4169 \eql@define@key\eql@keycat{linewidth,width}{\def\eql@display@linewidth{#1}}

```

**Horizontal Spacing and Columns.** Configure column spacing and compression threshold:

```

4170 \def\eql@keycat{equations,setup}
4171 \eql@define@key\eql@keycat{alignshrink}{\eql@decide@select{#3}{#2}{#1}{%
4172   {{max,full}{\eql@alignbadness@inf@bad}},%

```

```

4173    {{high}}{\eq@alignbadness@54\relax}},%
4174    {{med,medium}}{\eq@alignbadness@18\relax}},%
4175    {{low}}{\eq@alignbadness@6\relax}},%
4176    {\eq@decide@false{\eq@alignbadness@z@}}}%
4177 \eq@define@key\eq@keycat{tagshrink}{\eq@decide@select{#3}{#2}{#1}{%
4178    {{max,full}}{\eq@tagbadness@inf@bad}},%
4179    {{high}}{\eq@tagbadness@54\relax}},%
4180    {{med,medium}}{\eq@tagbadness@18\relax}},%
4181    {{low}}{\eq@tagbadness@6\relax}},%
4182    {{max,full}}{\eq@tagbadness@inf@bad}},%
4183    {\eq@decide@false{\eq@tagbadness@z@}}}%
4184 \eq@define@key\eq@keycat{alignbadness}{\eq@alignbadness@numexpr#1\relax}
4185 \eq@define@key\eq@keycat{tagbadness}{\eq@tagbadness@numexpr#1\relax}
4186 \eq@define@key\eq@keycat{mincolsep}{\eq@decide@select{#3}{#2}{#1}{%
4187    {\eq@decide@false{\def\eq@colsepmin@val{0pt}}},%
4188    {\relax{\def\eq@colsepmin@val{#1}}}}}%
4189 \eq@define@key\eq@keycat{maxcolsep}{\eq@decide@select{#3}{#2}{#1}{%
4190    {\eq@decide@false{\def\eq@colsepmax@val{.5\maxdimen}}},%
4191    {\relax{\def\eq@colsepmax@val{#1}}}}}%
4192 \eq@define@key\eq@keycat{fulllength}[true]{%
4193    \eq@decide@bool{#3}{#2}{#1}\eq@columns@fulllength}

4194 \eq@define@key{equationsbox,setup}{colsep}{\def\eq@box@colsep{#1}}

```

**Horizontal Shape.** Configure horizontal alignment schemes:

```

4195 \def\eq@keycat{equations,equationsbox,setup}
4196 \eq@define@key\eq@keycat{shape}[default]{\eq@shape@set{#1}}
4197 \eq@define@key\eq@keycat{padding,pad}[indent]{%
4198    \eq@decide@select{#3}{#2}{#1}{%
4199      {{max}}{\let\eq@paddingleft@val\@undefined}},%
4200      {{indent}}{\def\eq@paddingleft@val{\eq@indent@val}}},%
4201      {\eq@decide@false{\def\eq@paddingleft@val{0pt}}},%
4202      {\relax{\def\eq@paddingleft@val{#1}}}}%
4203    \let\eq@paddingright@val\eq@paddingleft@val}
4204 \eq@define@key\eq@keycat{padleft}[indent]{%
4205    \eq@decide@select{#3}{#2}{#1}{%
4206      {{max}}{\let\eq@paddingleft@val\@undefined}},%
4207      {{indent}}{\def\eq@paddingleft@val{\eq@indent@val}}},%
4208      {\eq@decide@false{\def\eq@paddingleft@val{0pt}}},%
4209      {\relax{\def\eq@paddingleft@val{#1}}}}%
4210 \eq@define@key\eq@keycat{padright}[indent]{%
4211    \eq@decide@select{#3}{#2}{#1}{%
4212      {{max}}{\let\eq@paddingright@val\@undefined}},%
4213      {{indent}}{\def\eq@paddingright@val{\eq@indent@val}}},%
4214      {\eq@decide@false{\def\eq@paddingright@val{0pt}}},%
4215      {\relax{\def\eq@paddingright@val{#1}}}}%
4216 \eq@define@key\eq@keycat{indent}[2em]{%
4217    \def\eq@indent@val{#1}}

```

**Math Classes at Alignment.** Configure math classes at alignment marker:

```

4218 \def\eq@keycat{equations,equationsbox,setup}
4219 \eq@define@key\eq@keycat{classout}{\eq@class@innerleft@set{#1}}
4220 \eq@define@key\eq@keycat{classin}{\eq@class@innerright@set{#1}}
4221 \eq@define@key\eq@keycat{classlead,classin*}{\eq@class@innerlead@set{#1}}
4222 \eq@define@key\eq@keycat{ampeq}[]{\eq@class@ampeq}
4223 \eq@define@key\eq@keycat{eqamp}[]{\eq@class@eqamp}

```

```

4224 \eqld@define@key\eqld@keycat{class}{\eqld@decide@select{#3}{#2}{#1}}{%
4225   {{\ampeq,amprel,eqafter,beforerel}\eqld@class@ampeq},%
4226   {{eqamp,relamp,eqbefore,afterrel}\eqld@class@eqamp}}}

```

**Punctuation.** Configure punctuation defaults:

```

4227 \let\eqld@punct@main\relax
4228 \def\eqld@keycat{equations,equationsbox,setup}
4229 \eqld@define@key\eqld@keycat{punctsep}{\,}{\def\eqld@punct@sep{#1}}
4230 \eqld@define@key\eqld@keycat{punct}{.}{\def\eqld@punct@main{#1}}
4231 \eqld@define@key\eqld@keycat{punctline}{,}{\def\eqld@punct@line{#1}}
4232 \eqld@define@key\eqld@keycat{punctcol}{,}{\def\eqld@punct@col{#1}}
4233 \eqld@define@key\eqld@keycat{punct*}[]{\let\eqld@punct@main\relax}
4234 \eqld@define@key\eqld@keycat{punctline*}[]{\let\eqld@punct@line\relax}
4235 \eqld@define@key\eqld@keycat{punctcol*}[]{\let\eqld@punct@col\relax}

```

**Alternative Content Description.** Alternative content description for accessibility or documentation purposes: **TODO:** implement in pdf tagging

```

4236 \eqld@define@key{equations,equationsbox}{alt}{}

```

**Global Switches.** Set global switches:

```

4237 \let\eqld@multi@linesfallback\eqld@false
4238 \let\eqld@scan@par\eqld@false
4239 \let\eqld@single@crerror\eqld@false
4240 \let\eqld@ampproof@active\eqld@false

4241 \eqld@define@key{setup}{linesfallback}[true]{%
4242   \eqld@decide@select{#3}{#2}{#1}}{%
4243     {\eqld@decide@false{\let\eqld@multi@linesfallback\eqld@false}},%
4244     {\eqld@decide@true{\let\eqld@multi@linesfallback\eqld@true}},%
4245     {{reuse,lean}{\let\eqld@multi@linesfallback\z@}},%
4246     {{measure,full}{\let\eqld@multi@linesfallback\eqld@true}}}}
4247 \eqld@define@key{setup}{ampproof}[true]{%
4248   \eqld@decide@bool{#3}{#2}{#1}\eqld@ampproof@active}
4249 \eqld@define@key{setup}{crerror}[true]{%
4250   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@crerror}
4251 \eqld@define@key{equations,setup}{rescan}[true]{%
4252   \eqld@decide@if{#3}{#2}{#1}%
4253   {\let\eqld@scan@body\eqld@scan@body@rescan}%
4254   {\let\eqld@scan@body\eqld@scan@body@dump}}
4255 \eqld@define@key{equations,equationsbox,setup}{scanpar}[true]{%
4256   \eqld@decide@bool{#3}{#2}{#1}\eqld@scan@par}
4257 \eqld@define@key{setup}{defaults}{%
4258   \eqld@decide@select{#3}{#2}{#1}}{%
4259     {{classic}{\eqld@defaults@classic}},%
4260     {{eqnlines}{\eqld@defaults@eqnlines}}}}

```

**Package Options.** Declare choices available at loading of package only:

```

4261 \let\eqld@provide@opt@equation\eqld@true
4262 \let\eqld@provide@opt@amsmathends\eqld@true
4263 \let\eqld@provide@opt@backup\eqld@false
4264 \let\eqld@provide@opt@amsmath\eqld@true
4265 \let\eqld@provide@opt@ang\eqld@true
4266 \let\eqld@provide@opt@eqref\eqld@true

```

```

4267 \eql@define@key{setup}{equation}[true]{%
4268   \eql@error@packageoption{#2}%
4269   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@equation}
4270 \eql@define@key{setup}{amsmathends}[true]{%
4271   \eql@error@packageoption{#2}%
4272   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
4273 \eql@define@key{setup}{backup}[true]{%
4274   \eql@error@packageoption{#2}%
4275   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
4276 \eql@define@key{setup}{amsmath}[true]{%
4277   \eql@error@packageoption{#2}%
4278   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmath}
4279 \eql@define@key{setup}{ang}[true]{%
4280   \eql@error@packageoption{#2}%
4281   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
4282 \eql@define@key{setup}{eqref}[true]{%
4283   \eql@error@packageoption{#2}%
4284   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

## P.4 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  $\TeX$ ,  $\LaTeX$  and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

4285 \def\eql@defaults@classic{%
4286   \eqnlineset{numberline=all}%
4287   \eqnlineset{mintagsep={.5\fontdimen6\textfont2\relax}}%
4288   \eqnlineset{maxcolsep=off}%
4289   \eqnlineset{spread={\jot}}%
4290   \eqnlineset{tagmargin}%
4291   \eqnlineset{tagmarginratio=1}%
4292   \eqnlineset{tagmarginthreshold=0.5}%
4293   \eqnlineset{leftmargin={\leftmargini}}%
4294   \eqnlineset{padding=max}%
4295   \eqnlineset{bestlineauto=off}%
4296   \eqnlineset{displayheight=off}%
4297   \eqnlineset{displaydepth=off}%
4298   \eqnlineset{shortmode=belowsingle}%
4299   \eqnlineset{abovecontmode=short}%
4300   \eqnlineset{belowcontmode=short}%
4301   \eqnlineset{aboveparmode=long}%
4302   \eqnlineset{belowparmode=long}%
4303   \eqnlineset{abovetopmode=long}%
4304   \eqnlineset{belowtopmode=long}%
4305   \eqnlineset{abovelongskip={\abovedisplayskip}}%
4306   \eqnlineset{belowlongskip={\belowdisplayskip}}%
4307   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
4308   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
4309   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
4310   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
4311   \eqnlineset{abovecontskip=0pt}%
4312   \eqnlineset{belowcontskip=0pt}%
4313   \eqnlineset{aboveparskip=0pt}%

```



```

4314 \eqnlineset{belowparskip=0pt}%
4315 \eqnlineset{abovetopskip=0pt}%
4316 \eqnlineset{belowtopskip=0pt}%
4317 \eqnlineset{abovetagskip=0pt}%
4318 \eqnlineset{belowtagskip=0pt}%
4319 \eqnlineset{abovemedtagskip=0pt}%
4320 \eqnlineset{belowmedtagskip=0pt}%
4321 \eqnlineset{abovepartagskip=0pt}%
4322 \eqnlineset{belowpartagskip=0pt}%
4323 \eqnlineset{crerror=false}%
4324 \eqnlineset{linesfallback=false}%
4325 }

```

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):

```

4326 \def\eq1@defaults@eqnlines{%
4327   \eqnlineset{numberline=all}%
4328   \eqnlineset{mintagsep=.5em}%
4329   \eqnlineset{maxcolsep=2em}%
4330   \eqnlineset{spread={0.2\normalbaselineskip}}%
4331   \eqnlineset{tagmargin}%
4332   \eqnlineset{tagmarginratio=.334}%
4333   \eqnlineset{tagmarginthreshold=0.5}%
4334   \eqnlineset{leftmargin={\leftmargini}}%
4335   \eqnlineset{padding=0pt}%
4336   \eqnlineset{bestlineauto}%
4337   \eqnlineset{displayheight=strut}%
4338   \eqnlineset{displaydepth=strut}%
4339   \eqnlineset{shortmode=above}%
4340   \eqnlineset{abovecontmode=noskip}%
4341   \eqnlineset{belowcontmode=long}%
4342   \eqnlineset{aboveparmode=long}%
4343   \eqnlineset{belowparmode=long}%
4344   \eqnlineset{abovetopmode=noskip}%
4345   \eqnlineset{belowtopmode=long}%
4346   \eqnlineset{longskip={0.75\normalbaselineskip
4347     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
4348   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
4349     plus 0.25\normalbaselineskip}}%
4350   \eqnlineset{belowshortskip={0.0\normalbaselineskip
4351     plus 0.25\normalbaselineskip}}%
4352   \eqnlineset{medskip={0.4\normalbaselineskip
4353     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
4354   \eqnlineset{abovecontskip=0pt}%
4355   \eqnlineset{belowcontskip=0pt}%
4356   \eqnlineset{aboveparskip=0pt}%
4357   \eqnlineset{belowparskip=0pt}%
4358   \eqnlineset{abovetopskip=0pt}%
4359   \eqnlineset{belowtopskip=0pt}%
4360   \eqnlineset{abovetagskip=0pt}%
4361   \eqnlineset{belowtagskip=0pt}%
4362   \eqnlineset{abovemedtagskip=0pt}%

```

```

4363 \eqnlineset{belowmedtagskip=0pt}%
4364 \eqnlineset{abovepartagskip=0pt}%
4365 \eqnlineset{belowpartagskip=0pt}%
4366 \eqnlineset{crerror=true}%
4367 \eqnlineset{linesfallback=true}%
4368 }

```

## P.5 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@movestar
@provide@undefinecmd
@provide@undefineenv
4369 \def\eql@provide@movecmd#1#2{%
4370   \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
4371 }
4372 \def\eql@provide@moveenv#1#2{%
4373   \eql@provide@movecmd{#1}{#2}%
4374   \ifcsname end#2\endcsname
4375     \eql@provide@movecmd{end#1}{end#2}%
4376   \fi
4377 }
4378 \def\eql@provide@movestar#1#2{%
4379   \eql@provide@moveenv{#1}{#2}%
4380   \ifcsname #2*\endcsname
4381     \eql@provide@moveenv{#1*}{#2*}%
4382   \fi
4383 }
4384 \def\eql@provide@undefinecmd#1{%
4385   \expandafter\let\csname #1\endcsname\@undefined
4386 }
4387 \def\eql@provide@undefineenv#1{%
4388   \eql@provide@undefinecmd{#1}%
4389   \eql@provide@undefinecmd{end#1}%
4390 }

```

**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:

```

4391 \def\eql@amsmath@endfix#1#2{%
4392   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
4393   \expandafter\ifx\csname end#1\endcsname\@tempa
4394     \eql@provide@movecmd{end#1}{end#2}%
4395   \fi
4396 }

```

`\eql@amsmath@fixends` Perform the replacement for all `amsmath` environments whenever `amsmath` is loaded:

```

4397 \def\eql@amsmath@fixends{%
4398   \eql@amsmath@after{%
4399     \eql@amsmath@endfix{gather*}{gather}%
4400     \eql@amsmath@endfix{multline*}{multline}%
4401     \eql@amsmath@endfix{align*}{align}%
4402     \eql@amsmath@endfix{flalign*}{align}%
4403     \eql@amsmath@endfix{flalign*}{align}%
4404     \eql@amsmath@endfix{alignat}{align}%
4405     \eql@amsmath@endfix{alignat*}{align}%
4406     \eql@amsmath@endfix{xalignat}{align}%
4407     \eql@amsmath@endfix{xalignat*}{align}%
4408     \eql@amsmath@endfix{xxalignat}{align}%
4409     \eql@amsmath@endfix{gathered}{aligned}%
4410     \eql@amsmath@endfix{alignedat}{aligned}%
4411   }
4412 }
```

**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:

```

4413 \def\eql@provide@backup@amsmath#1{%
4414   \eql@amsmath@after{%
4415     \eql@provide@moveenv{ams#1}{#1}%
4416   }%
4417 }
```

`\provide@backup@eqref` Copy an `eqref` to `amseqref` whenever `amsmath` is loaded:

```

4418 \def\eql@provide@backup@eqref{%
4419   \eql@amsmath@after{%
4420     \eql@provide@movecmd{amseqref}{eqref}%
4421   }%
4422 }
```

`\provide@backup@multlined` The environment `multlined` is supplied by `mathtools`. We copy it to `amsmultlined` anyway, but whenever `mathtools` is loaded:

```

4423 \def\eql@provide@backup@multlined{%
4424   \AddToHook{package/mathtools/after}{%
4425     \eql@provide@moveenv{amsmultlined}{multlined}%
4426   }%
4427 }
```

`\provide@backup@equation` The  $\text{\LaTeX}$  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  $\text{\LaTeX}$  and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original  $\text{\LaTeX}$  `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:

```

4428 \def\eql@provide@backup@equation{%
4429   \eql@amsmath@after{%
```

```

4430 \eql@provide@moveenv{amsequeation}{equation}%
4431 \eql@tagging@register@env{amsequeation}%
4432 \eql@provide@moveenv{amsequeation*}{equation*}%
4433 \eql@tagging@register@env{amsequeation*}%
4434 }%
4435 \AddToHook{package/hyperref/after}{%
4436 \@ifpackageloaded{amsmath}{}%{%
4437 \let\latexequation\H@equation
4438 \let\endlatexequation\H@endequation
4439 \eql@tagging@register@env{latexequation}%
4440 \eql@provide@moveenv{hyperrefequation}{equation}%
4441 \eql@tagging@register@env{hyperrefequation}%
4442 }%
4443 }%
4444 \@ifpackageloaded{amsmath}{%}{\@ifpackageloaded{hyperref}{%}{%
4445 \eql@provide@moveenv{latexequation}{equation}%
4446 \eql@tagging@register@env{latexequation}%
4447 }}%
4448 }

```

e@backup@displaymath **TODO:** describe

```

4449 \def\eql@provide@backup@displaymath{%
4450 \eql@provide@moveenv{latexdisplaymath}{displaymath}%
4451 }

```

@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:

```

4452 \def\eql@provide@backup@subequations{%
4453 \eql@amsmath@after{%
4454 \eql@provide@moveenv{amssubequations}{subequations}%
4455 }%
4456 \AddToHook{package/hyperref/after}{%
4457 \AddToHook{cmd/amssubequations/before}%
4458 {%
4459 \stepcounter{equation}%
4460 \protected@edef\theHparentequation{\theHequation}%
4461 \addtocounter{equation}{-1}%
4462 }%
4463 \AddToHook{cmd/amssubequations/after}%
4464 {%
4465 \def\theHequation{\theHparentequation\alph{equation}}%
4466 \ignorespaces
4467 }%
4468 }%
4469 }

```

\eql@provide@backup Backup all amsmath environments:

```

4470 \def\eql@provide@backup{%
4471 \eql@provide@backup@eqref
4472 \eql@provide@backup@equation
4473 \eql@provide@backup@displaymath
4474 \eql@provide@backup@amsmath{gather}%
4475 \eql@provide@backup@amsmath{gather*}%
4476 \eql@provide@backup@amsmath{multline}%
4477 \eql@provide@backup@amsmath{multline*}%

```

```

4478 \eql@provide@backup@amsmath{align}%
4479 \eql@provide@backup@amsmath{align*}%
4480 \eql@provide@backup@amsmath{flalign}%
4481 \eql@provide@backup@amsmath{flalign*}%
4482 \eql@provide@backup@amsmath{alignat}%
4483 \eql@provide@backup@amsmath{alignat*}%
4484 \eql@provide@backup@amsmath{xalignat}%
4485 \eql@provide@backup@amsmath{xalignat*}%
4486 \eql@provide@backup@amsmath{xxalignat}%
4487 \eql@provide@backup@amsmath{aligned}%
4488 \eql@provide@backup@amsmath{aligned*}%
4489 \eql@provide@backup@amsmath{alignedat}%
4490 \eql@provide@backup@amsmath{alignedat*}%
4491 \eql@provide@backup@amsmath{gathered}%
4492 \eql@provide@backup@amsmath{gathered*}%
4493 \eql@provide@backup@multlined
4494 \eql@provide@backup@subequations
4495 }

```

### Replacement amsmath Environments.

`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.)
4496 \newenvironment{eql@gathered}
4497 {\eqnaddopt{lines}\equationsbox}\endequationsbox}
4498 \newenvironment{eql@multlined}
4499 {\eqnaddopt{lines,padding,shape=steps}\equationsbox}\endequationsbox}
4500 \newenvironment{eql@aligned}
4501 {\eqnaddopt{align}\equationsbox}\endequationsbox}
4502 \newenvironment{eql@alignedat}[1]
4503 {\eqnaddopt{align,colsep=0pt}\equationsbox}\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.*)

```

4504 \newenvironment{eql@equation}
4505 {\eqnaddopt{equation}\equations}\endequations}
4506 \newenvironment{eql@displaymath}
4507 {\eqnaddopt{equation,number}\equations}\endequations}
4508 \newenvironment{eql@gather}
4509 {\eqnaddopt{lines}\equations}\endequations}
4510 \newenvironment{eql@multline}
4511 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
4512 {\endequations}
4513 \newenvironment{eql@align}
4514 {\eqnaddopt{align}\equations}\endequations}
4515 \newenvironment{eql@flalign}
4516 {\eqnaddopt{align,fulllength}\equations}\endequations}
4517 \newenvironment{eql@alignat}
4518 {\eqnaddopt{mincolsep=0pt,maxcolsep=0pt}\eql@xalignat}\endequations}
4519 \newenvironment{eql@xalignat}[1]
4520 {\eql@align}\endequations}
4521 \newenvironment{eql@xxalignat}[1]
4522 {\eql@flalign}\endequations}
4523 \newenvironment{eql@equation*}
4524 {\eqnaddopt{nonumber}\eql@equation}\endequations}

```

```

4525 \newenvironment{eql@gather*}
4526   {\eqnaddopt{nonumber}\eql@gather}{\endequations}
4527 \newenvironment{eql@multline*}
4528   {\eqnaddopt{nonumber}\eql@multline}{\endequations}
4529 \newenvironment{eql@align*}
4530   {\eqnaddopt{nonumber}\eql@align}{\endequations}
4531 \newenvironment{eql@flalign*}
4532   {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
4533 \newenvironment{eql@alignat*}
4534   {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
4535 \newenvironment{eql@xalignat*}
4536   {\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):

```

4537 \def\eql@provide@onlyonce#1#2{%
4538   \def\eql@tmp{#2}%
4539   \def\@tempa{#1}%
4540   \ifx\eql@tmp\@tempa
4541     \let\eql@tmp\@empty
4542   \fi
4543   \ifx\eql@tmp\@empty
4544     \let\eql@tmp\@undefined
4545     \ifx\@nodocument\relax
4546       \def\eql@tmp{#1}%
4547     \fi
4548     \ifcsname eql@provided@#1\endcsname
4549       \def\eql@tmp{#1}%
4550     \fi
4551     \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4552   \else
4553     \def\@tempa{*}%
4554     \ifx\eql@tmp\@tempa
4555       \def\eql@tmp{#1}%
4556     \fi
4557   \fi
4558 }

```

**\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:

```

4559 \def\eql@provide@eqref#1{%
4560   \def\eql@tmp{#1}%
4561   \def\@tempa{\eqref}%
4562   \ifx\eql@tmp\@tempa
4563     \let\eql@tmp\@empty
4564   \fi
4565   \ifx\eql@tmp\@empty

```

```

4566 \eql@provide@onlyonce{eqref}{}%
4567 \else
4568 \def\@tempa{*}%
4569 \ifx\eql@tmp\@tempa
4570 \def\eql@tmp{eqref}%
4571 \else
4572 \edef\eql@tmp{\expandafter\@gobble\string#1}%
4573 \fi
4574 \fi
4575 \ifdefined\eql@tmp
4576 \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
4577 \else
4578 \eql@amsmath@after{%
4579 \eql@provide@movecmd{eqref}{eql@eqref}%
4580 }%
4581 \AddToHook{package/mathtools/after}{%
4582 \eql@provide@movecmd{eqref}{eql@eqref}%
4583 }%
4584 \eql@provide@movecmd{eqref}{eql@eqref}%
4585 \eql@amsmath@undefine\eqref
4586 \fi
4587 }

```

`\eql@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:

```

4588 \def\eql@provide@amsmath#1#2{%
4589 \eql@provide@onlyonce{#1}{#2}%
4590 \ifdefined\eql@tmp
4591 \expandafter\eql@provide@movestart\expandafter{\eql@tmp}{eql@#1}%
4592 \else
4593 \eql@amsmath@after{%
4594 \eql@provide@movestart{#1}{eql@#1}%
4595 }%
4596 \AddToHook{package/mathtools/after}{%
4597 \eql@provide@movestart{#1}{eql@#1}%
4598 }%
4599 \eql@provide@movestart{#1}{eql@#1}%
4600 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
4601 \ifcsname eql@#1*\endcsname
4602 \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
4603 \fi
4604 \fi
4605 }

```

`\eql@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:

```

4606 \def\eql@provide@multlined#1{%
4607 \eql@provide@onlyonce{multlined}{#1}%
4608 \ifdefined\eql@tmp
4609 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
4610 \else
4611 \AddToHook{package/mathtools/after}{%
4612 \eql@provide@moveenv{multlined}{eql@multlined}%
4613 }%
4614 \eql@provide@moveenv{multlined}{eql@multlined}%

```

```

4615 \ifpackageloaded{mathtools}{}{\AddToHook{package/mathtools/before}{%
4616 \eq@provide@undefineenv{multlined}}}%
4617 \fi
4618 }

```

**eq@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:

```

4619 \def\eq@provide@equation#1{%
4620 \eq@provide@onlyonce{equation}{#1}%
4621 \ifdefined\eq@tmp
4622 \expandafter\eq@provide@movestart\expandafter{\eq@tmp}{eq@equation}%
4623 \else
4624 \eq@amsmath@after{%
4625 \eq@provide@movestart{equation}{eq@equation}%
4626 }%
4627 \AddToHook{package/hyperref/after}{%
4628 \ifpackageloaded{amsmath}{}{%
4629 \eq@provide@moveenv{equation}{eq@equation}%
4630 }%
4631 }%
4632 \eq@provide@movestart{equation}{eq@equation}%
4633 \eq@amsmath@before{\eq@provide@undefineenv{equation*}}%
4634 \ifdefined\eq@tagging@on
4635 \AddToHook{begindocument/end}{%
4636 \eq@provide@movestart{equation}{eq@equation}%
4637 }%
4638 \fi
4639 \fi
4640 }

```

**@provide@displaymath** **TODO:** describe

```

4641 \def\eq@provide@displaymath#1{%
4642 \eq@provide@onlyonce{displaymath}{#1}%
4643 \ifdefined\eq@tmp
4644 \expandafter\eq@provide@moveenv\expandafter{\eq@tmp}{eq@displaymath}%
4645 \else
4646 \eq@provide@moveenv{displaymath}{eq@displaymath}%
4647 \ifdefined\eq@tagging@on
4648 \AddToHook{begindocument/end}{%
4649 \eq@provide@moveenv{displaymath}{eq@displaymath}%
4650 }%
4651 \fi
4652 \fi
4653 }

```

**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 empty). 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:

```

4654 \def\eq@provide@subequations#1{%
4655 \eq@provide@onlyonce{subequations}{#1}%

```



```

4656 \ifdefined\eql@tmp
4657   \expandafter\eql@provide@moveenv
4658   \expandafter{\eql@tmp}{eql@subequations}%
4659 \else
4660   \eql@amsmath@after{%
4661     \eql@provide@moveenv{subequations}{eql@subequations}%
4662   }%
4663   \AddToHook{package/hyperref/after}{%
4664     \AddToHook{cmd/subequations/before}[hyperref]{}%
4665     \AddToHook{cmd/subequations/after}[hyperref]{}%
4666     \RemoveFromHook{cmd/subequations/before}[hyperref]%
4667     \RemoveFromHook{cmd/subequations/after}[hyperref]%
4668     \AddToHook{begindocument/end}{%
4669       \eql@provide@moveenv{subequations}{eql@subequations}%
4670     }%
4671   }%
4672   \eql@provide@moveenv{subequations}{eql@subequations}%
4673   \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
4674 \fi
4675 }

```

`\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:

```

4676 \def\eql@provide@sqr{%
4677   \let\[\eql@sqr@open
4678   \let\]\eql@sqr@close
4679   \eql@amsmath@after{%
4680     \let\[\eql@sqr@open
4681     \let\]\eql@sqr@close
4682   }%
4683   \ifdefined\eql@tagging@on
4684     \AddToHook{begindocument/end}{%
4685       \let\[\eql@sqr@open
4686       \let\]\eql@sqr@close
4687     }%
4688   \fi
4689 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```

4690 \def\eql@provide@ang{%
4691   \let\<\eql@ang@open
4692   \let\>\eql@ang@close
4693 }

```

## Interface.

**provide** (*key*) We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

4694 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
4695 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
4696 \eql@define@key{provide}{gather}[]{\eql@provide@amsmath{gather}{#1}}
4697 \eql@define@key{provide}{multline}[]{\eql@provide@amsmath{multline}{#1}}
4698 \eql@define@key{provide}{align}[]{\eql@provide@amsmath{align}{#1}}

```

```

4699 \eqld@define@key{provide}{flalign}[]{\eqld@provide@amsmath{flalign}{#1}}
4700 \eqld@define@key{provide}{alignat}[]{\eqld@provide@amsmath{alignat}{#1}}
4701 \eqld@define@key{provide}{xalignat}[]{\eqld@provide@amsmath{xalignat}{#1}}
4702 \eqld@define@key{provide}{xxalignat}[]{\eqld@provide@amsmath{xxalignat}{#1}}
4703 \eqld@define@key{provide}{aligned}[]{\eqld@provide@amsmath{aligned}{#1}}
4704 \eqld@define@key{provide}{alignedat}[]{\eqld@provide@amsmath{alignedat}{#1}}
4705 \eqld@define@key{provide}{gathered}[]{\eqld@provide@amsmath{gathered}{#1}}
4706 \eqld@define@key{provide}{multlined}[]{\eqld@provide@multlined{#1}}
4707 \eqld@define@key{provide}{subequations}[]{\eqld@provide@subequations{#1}}
4708 \eqld@define@key{provide}{sqr}[]{\eqld@provide@sqr}
4709 \eqld@define@key{provide}{ang}[]{\eqld@provide@ang}
4710 \eqld@define@key{provide}{eqref}[]{\eqld@provide@eqref{#1}}
4711 \eqld@define@key{provide}{tagform}[]{%
4712   \def\tagform@##1{\maketag@@@{\eqld@tag@form{#1}}}}
4713 \eqld@define@key{provide}{maketag}[]{%
4714   \def\maketag@@@##1{\eqld@tag@box{##1}}}

```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```

4715 \newcommand{\eqnlinesprovide}[1]{%
4716 (dev)\eqld@dev@start\eqnlinesprovide
4717   \eqld@setkeys{provide}{#1}%
4718   \ignorespaces
4719 }

```

## P.6 Global and Package Options

Handle global and package options:

`\eqnlinesset` The macro `\eqnlinesset` processes global configuration options including the package options:

```

4720 \newcommand{\eqnlinesset}[1]{%
4721 (dev)\eqld@dev@start\eqnlinesset
4722   \eqld@setkeys{setup}{#1}%
4723   \ignorespaces
4724 }

```

Disable error message for exclusive package options:

```
4725 \let\eqld@error@packageoption\gobble
```

Declare math layout options `leqno` and `fleqn` for common L<sup>A</sup>T<sub>E</sub>X classes:

```

4726 \DeclareOption{leqno}{\eqnlinesset{tagsleft}}
4727 \DeclareOption{fleqn}{\eqnlinesset{left}}

```

Pass undeclared options on to keyval processing:

```
4728 \DeclareOption*{\expandafter\eqnlinesset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

4729 \eqld@defaults@eqnlines
4730 \eqld@mode@columns
4731 \eqld@mode@aligned

```

Process package options:

```
4732 \ProcessOptions
```

`@error@packageoption` Enable error message for exclusive package options:

```
4733 \def\eql@error@packageoption#1{%
4734   \eql@error{may only use ‘#1’ as a package option}%
4735 }
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the  $\TeX$  conditional parsing mechanism:

```
4736 \ifdefined\tagsleft@true\else
4737   \expandafter\newif\csname iftagsleft@\endcsname
4738 \fi
4739 \ifdefined\@fleqntrue\else
4740   \expandafter\newif\csname if@fleqn\endcsname
4741 \fi
```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`:

```
4742 \ifdefined\eql@provide@opt@amsmath
4743   \let\eql@provide@opt@equation\eql@true
4744   \eql@amsmath@after{%
4745     \iftagsleft@
4746       \eqnlineset{tagsleft}
4747     \else
4748       \eqnlineset{tagsright}
4749     \fi
4750     \if@fleqn
4751       \eqnlineset{left}
4752     \else
4753       \eqnlineset{center}
4754     \fi
4755   }
4756 \fi
```

Make the ending statements for `amsmath` environments independent if desired, so that they may be overwritten individually:

```
4757 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

Backup all `amsmath` environments that may be overwritten to `ams...`. This will happen before any replacements:

```
4758 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

Provide native  $\LaTeX$  environment `equation` and symbolic shortcut `\[...\]` if desired:

```
4759 \ifdefined\eql@provide@opt@equation\eqnlinesprovide{%
4760   equation,sqr,displaymath}
4761 \fi
```

Provide `amsmath` equation environments if desired:

```
4762 \ifdefined\eql@provide@opt@amsmath
4763   \eqnlinesprovide{%
4764     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
4765     multlined,gathered,aligned,alignedat,%
4766     subequations}
4767 \fi
```

Provide symbolic shortcut `\<...\>` if desired:

```
4768 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
4769 \ifdefined\eq\provideopt@eqref\eqnlinesprovide{eqref}\fi
```