

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`

2025/04/03, v0.7

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

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 \LaTeX and the `amsmath` package with many options for convenient adjustment of the intended layout. In particular, the package adds flexible schemes for numbering, horizontal alignment and semi-automatic punctuation, and it improves upon the horizontal and vertical spacing options. The extensions can be used and adjusted through optional arguments and modifiers to the equation environments as well as global settings.

Contents

1	Introduction	3
2	Usage	4
2.1	Equations Environment	5
2.2	Numbering	7
2.3	Horizontal Adjustment	10
2.4	Punctuation	13
2.5	Math Classes at Alignment	14
2.6	Vertical Spacing	15
2.7	Further Environments	18
2.8	General Options	19
2.9	Feature Selection and Package Options	19
3	Information	20
3.1	Copyright	20
3.2	Credits	20
3.3	Files and Installation	20
3.4	Related CTAN Packages	21
3.5	Feature Suggestions	21
3.6	Revision History	22
A	Implementation	22

B	General Support	22
B.1	Development Messages	23
B.2	Supporting Definitions	23
B.3	Dollardollar Abstraction	23
B.4	Look-Ahead in Alignment	23
B.5	Error Messages	25
B.6	amsmath Integration	26
B.7	PDF Tagging Support	26
C	Parameters and Registers	27
C.1	Parameters	27
C.2	Registers	31
C.3	Hooks	32
D	Punctuation	33
E	Math Classes at Alignment	34
F	Equation Numbering	35
F.1	Tag Formatting	35
F.2	Showkeys Integration	36
F.3	Labels	37
F.4	Tags	37
F.5	Anchors	38
F.6	Tag Composition	39
F.7	Tagbox Methods	40
F.8	Numbering Schemes	42
F.9	Numbering Framework	44
G	Subequation Numbering	48
G.1	Definitions	48
G.2	Environment	49
G.3	Subequation Scheme	50
H	Display Equations Support	51
H.1	Display Breaks	51
H.2	General Initialisation	52
H.3	halign Support	56
H.4	Stack	57
I	Horizontal Spacing for Lines	59
I.1	Supporting Definitionss	59
I.2	Shape Schemes	60
I.3	Adjustment Methods	62
I.4	Central Alignment Layout	64
I.5	Left Alignment Layout	69
J	Multi-Line Support	73
J.1	Table Registers	73
J.2	Measure Support	74
J.3	Print Support	75
J.4	Line Breaks	75
J.5	Intertext	76
K	Equations Box Environment	78

K.1	Line Breaks	78
K.2	Stacked Mode	78
K.3	Aligned Mode	80
K.4	Main	81
K.5	Environment	82
L	Single-Line Equation	82
L.1	Native Mode	82
L.2	Print	83
M	Multi-Line Lines Mode	84
M.1	Measure	84
M.2	Print	85
N	Multi-Line Columns Mode	87
N.1	Columns Processing	87
N.2	Measure	87
N.3	Print	90
N.4	Adjust	93
O	Interface	99
O.1	Scanning the Equation Body	99
O.2	Options Processing	102
O.3	Single-Line Main	103
O.4	Multi-Line Main	104
O.5	Equations Environment	105
P	Options	108
P.1	Selection Tools	108
P.2	Declaration Code	109
P.3	Options Declarations	110
P.4	Parameter Sets	116
P.5	Component Selection	118
P.6	Global and Package Options	123

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 \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 `\langle...\rangle` as an alias multi-line equations.
- The horizontal alignment and indentation of equation lines can be adjusted via a scheme or on a line-by-line basis.
- The alignment marker can be placed before or after the equation signs while maintaining proper spacing to symbols before and after it. This simplifies the construction of continuing equations in an aligned context.
- Equation lines are subject to shrinking of space if the available space does not suffice (analogously to single-line equations).
- Most settings can be controlled via optional arguments and modifiers to the equation environment or via global settings. This includes switching between different types of equation environments, enabling or disabling numbering, adjusting vertical spacing, etc. This feature simplifies the adjustment and fine-tuning of equations towards the intended layout.
- Last but not least, the underlying `amsmath` code, originating from the \TeX era and early \LaTeX years, has been redesigned with emphasis on clarity, readability, adjustability and maintainability (but at the cost of moderately higher resource consumption and moderately lower efficiency). Nevertheless, it remains $\LaTeX 2_{\epsilon}$ code.

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

2 Usage

Notice regarding package version v0.7: 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

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

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

2.1 Equations Environment

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

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

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

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

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

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

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

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

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

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:

`lines` (*key*)

`columns` (*key*)

¹Arguably, 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.

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>\<~...></code>	<code>\<...></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&=\cos\phi & \phi&=\arccos x \ \ \</code>	$x = \cos \phi$	$\phi = \arccos x$	(4)
<code>&=(z+z^{-1})/2 & &=-i\log z</code>	$= (z + z^{-1})/2$	$= -i \log z$	(5)
<code>\end{equations}</code>			

`\[...]` The package offers several alternative names for the same mode as well as a symbolic short `\<...>` form `\<...>` extending the L^AT_EX display equation form `\[...]` to multi-line equations. `~` (*key*) Here, the tilde ‘~’ in `\<~...>` is a modifier character which acts as a short form for the `spropt` (*key*) optional argument `lines` selecting the lines mode. Both short forms can be customised by `angopt` (*key*) setting default arguments via the global options `spropt={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>\<~</code>			
<code>x=\cos\phi \ \ \ \ \ \phi=\arccos x</code>		$x = \cos \phi$	
<code>\></code>		$\phi = \arccos x$	
<code>\<</code>			
<code>x&=\cos\phi & \phi&=\arccos x \ \ \</code>	$x = \cos \phi$	$\phi = \arccos x$	
<code>&=(z+z^{-1})/2 & &=-i\log z</code>	$= (z + z^{-1})/2$	$= -i \log z$	
<code>\></code>			
<code>\eqnlineset{spropt={donumber}}</code>			
<code>\[x=\cos\phi \]</code>		$x = \cos \phi$	(6)

`equation` (*env.*) The package also supplies or overwrites the amsmath environments `equation`, `multline`, `gather` (*env.*) `align` (*env.*)

gather, align and flalign including their starred variants (but neither the alignat alternative forms nor the split construction). It is possible to define further equation environments *env* with a predefined set of options *opts* using:

```

\renewenvironment{env}{\eqnadopt{opts}\equations}{\endequations}

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

```

2.2 Numbering

numberline (*key*) The package extends the established interface of L^AT_EX 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
none	n		all lines, preset off
all	a	individual lines	all lines
sub	s		subequations (a, b, c, ...)
first	f		first line
last	l		last line
middle	m	single number	middle line
out	o		last/first line for right/left tags
in	i		first/last line for right/left tags
here	h		line indicated by \numberhere

```

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

```

none:	all:	sub:
$x = \cos \phi$	$x = \cos \phi$ (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)

first:	(17)	middle:	(18)	last:	(19)
$x = \cos \phi$		$x = \cos \phi$		$x = \cos \phi$	
$= (z + z^{-1})/2$		$= (z + z^{-1})/2$		$= (z + z^{-1})/2$	
$\phi = \arccos x$		$\phi = \arccos x$		$\phi = \arccos x$	
$= -i \log z$		$= -i \log z$		$= -i \log z$	

`\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` (`\nn='*' or \dn='!'` for short) or by `\number=bool` with *bool* either `on` or `off` (among several alternative forms). Alternatively the number can be switched by using modifiers (which cannot be used in conjunction with optional arguments [...]):

`\[*_... \]` and `\[!_... \]`

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

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

`\numbernext`

`\numberhere` and `\numbernext`

The former macro overrides the position to the present line, the latter macro defers the number to the next line. For example, if an equation is broken into several lines one may use the combination `\numbernext \\\` to assign the number to the last line.

<code>\begin{equations}</code>		
<code>x &= \cos\phi \nonumber \\\</code>	$x = \cos \phi$	
<code>&= (z+z^{-1})/2 \\\</code>	$= (z + z^{-1})/2$	(20)
<code>\phi &= \arccos x \nonumber \\\</code>	$\phi = \arccos x$	
<code>&= -i\log z</code>	$= -i \log z$	(21)
<code>\end{equations}</code>		
<code>\begin{equations}*</code>		
<code>x &= \cos\phi \donumber \\\</code>	$x = \cos \phi$	(22)
<code>&= (z+z^{-1})/2 \\\</code>	$= (z + z^{-1})/2$	
<code>\phi &= \arccos x \donumber \\\</code>	$\phi = \arccos x$	(23)
<code>&= -i\log z</code>	$= -i \log z$	
<code>\end{equations}</code>		
<code>\eqnlineset{numberline=last}</code>	$x = \cos \phi$	
<code>\<! x &= \cos\phi \\\</code>	$\phi = \arccos x$	(24)
<code>\phi &= \arccos x \></code>		
<code>\eqnlineset{angopt=donumber}</code>	$x = \cos \phi$	
<code>\<* x &= \cos\phi \\\</code>	$\phi = \arccos x$	
<code>\phi &= \arccos x \></code>		
<code>\begin{equations}</code>		
<code>x &= \cos\phi \numbernext \\\</code>	$x = \cos \phi$	
<code>&= (z+z^{-1})/2 \\\</code>	$= (z + z^{-1})/2$	(25)
<code>\phi &= \arccos x \numbernext \\\</code>	$\phi = \arccos x$	
<code>&= -i\log z</code>	$= -i \log z$	(26)
<code>\end{equations}</code>		


```

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

```

$$\begin{aligned}
x &= \cos \phi \\
&= (z + z^{-1})/2 \\
\phi &= \arccos x \\
&= -i \log z
\end{aligned}
\tag{27}$$

```

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

```

$$\begin{aligned}
x &= \cos \phi \\
&= (z + z^{-1})/2 \\
\phi &= \arccos x \\
&= -i \log z
\end{aligned}
\tag{28}$$

`\label` Equation numbers can receive L^AT_EX labels as usual and they can be turned into manually `\tag` assigned tags using the established macros:

$$\backslash\text{label}\{label\} \quad \text{and} \quad \backslash\text{tag}[*]\{tag\}$$

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-
`@` (*key*) tional arguments):

$$\text{label}=\{label\}, \quad \text{tag}[*]=\{tag\}, \quad \backslash\@{label}\dots\backslash$$

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.

$$\backslash\text{eqref}\{label\}.$$

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

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

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

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.

```

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

```

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

[29]

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

```

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

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

tagsright (*key*) `tags=left` or `tagsright`, `tagsleft` for short.

tagsleft (*key*)

```

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

```

$$\begin{aligned}
x &= \cos \phi && (30) \\
&= (z + z^{-1})/2 && (31) \\
\phi &= \arccos x && (32) \\
&= -i \log z && (33)
\end{aligned}$$

tagmargin (*key*) In central alignment layout, one can impose a tag margin `tagmargin={dimen}` which allo-

tagmargin* (*key*) cates some space to the tag such that equation content is centred in the remaining horizontal

tagmarginratio (*key*) space. The margin can also be set to the width of some text by `tagmargin*={text}` or it can

be calculated as the maximum width of tags by `tagmargin` without parameter (default). The

option `tagmarginratio={ratio}` uses the tag margin only for equation blocks with a ratio

of tags to rows above the given (decimal) ratio (a value above 1 uses the tag margin only for

single equations with tags; default is 0.334). The option `tagmarginthreshold={threshold}`

uses the tag margin only if the ratio of spacings would be below the given (decimal) thresh-

old (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}
\leftmargin 4em \tag{38}
\leftmargin 8em \tag{39}
\leftmargin 12em \tag{40}
\leftmargin 16em \tag{41}

```



`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}
\leftmargin 4em \tag{38}
\leftmargin 8em \tag{39}
\leftmargin 12em \tag{40}
\leftmargin 16em \tag{41}

```

`margins` (*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 `margins=on`), but it can be eliminated so that the outer columns are pushed right to the margin (option `margins=off`). 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*` (which is required for `margins=off` to take effect).

```

\leftmargin 4em \tag{38}
\leftmargin 8em \tag{39}
\leftmargin 12em \tag{40}
\leftmargin 16em \tag{41}

```

```

\leftmargin 4em \tag{38}
\leftmargin 8em \tag{39}
\leftmargin 12em \tag{40}
\leftmargin 16em \tag{41}

```

```

\leftmargin 4em \tag{38}
\leftmargin 8em \tag{39}
\leftmargin 12em \tag{40}
\leftmargin 16em \tag{41}

```

```

\leftmargin 4em \tag{38}
\leftmargin 8em \tag{39}
\leftmargin 12em \tag{40}
\leftmargin 16em \tag{41}

```

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

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

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

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

`\shoveleft` The alignment preset can be adjusted for individual lines by the macros:
`\shovecenter` `\shoveleft[*!][dimen]`, `\shovecenter`, `\shoveright`
`\shoveright`
`indent (key)`

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 `\shoveleft` also accepts the modifiers `*` or `!` for indentation or hanging indentation by the standard indentation amount (`indent=2em`) or an optional argument `[indent]` specifying a variable amount of indentation.

`padding (key)` Note that (hanging) indentation requires to add some padding around the equations block via `padleft (key)` the optional argument `padding[padleft|padright][=dimen]` or `padmax` to extend padding `padright (key)` to the whole line. Note that `indent*={dimen}` sets the default indentation amount and the `padmax (key)` left padding at the same time.

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

```

\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}
\>

```

marginbadness (*key*) Finally, we note that within single and stacked equations, very long equations that do not fit the available horizontal space are subject to shrinking attempts. In other words, \TeX will attempt to shrink the glue contained in the equation line to make it fit. This shrinking can be controlled by the two parameters **marginbadness** and **maxbadness** accepting integer values. The former is used for trying to shrink onto certain horizontal margins which are otherwise reserved for tags; the latter is used for using the maximal horizontal space which also raises or lowers the equation tag if needed. 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+x \\
x+x+x+x+x+x+x+x+x+x+x+x
\>

```

$$x + x \tag{46}$$

$$x + x + x + x \tag{47}$$

$$x + x + x + x + x + x + x \tag{48}$$

$$x + x + x + x + x + x + x + x \tag{49}$$

$$x + x + x + x + x + x + x + x + x \tag{50}$$

$$x + x + x + x + x + x + x + x + x + x + x \tag{51}$$

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).

```

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

```

$$\phi = - \int \frac{dx}{\sqrt{1+x^2}} \tag{52}$$

$$x = \frac{\partial}{\partial \phi} \sin \phi \tag{53}$$

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.

The package supplies a semi-automatic scheme by which equations are terminated by a specific punctuation mark. ² Punctuation marks are set by:

²Clearly, 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.

```

\eqnpunctmain{punct}   \eqnlineset{punct={punct}}
\eqnpunct{punct}      \eqnaddopt{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 ‘`\eqnpunctmain.`’ 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.

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

`\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 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 columns and lines which are specified via the macros `\eqnpunctcol` and `\eqnpunctline` or `\eqnpunctline` the optional arguments `punctcol` and `punctline`. A punctuation item may also be handed on to the next lower level of punctuation via the starred forms `punct*` and `punctline*`.

```

\eqnpunctcol, \eqnpunctline;
\eqnpunctmain.
< 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_EX’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 \\\
  &= d-e \\\
  &\mathrel{ }\phantom{=} +f
\>
\<[class=eqamp]
a =& b+c \\\
  =& d-e \\\
  & +f
\>

```

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

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

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

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

2.6 Vertical Spacing

Display equations in $\text{T}_{\text{E}}\text{X}$ 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, $\text{T}_{\text{E}}\text{X}$ 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

`displayheight` (*key*) some potential gaps of the surrounding text. The apparent height and depth are defined by `displaydepth` (*key*) `displayheight` and `displaydepth` 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 T_EX feature for all single- and multi-line equation environments.

<p>example of a long text line: $\lbrack \mbox{long mode} \rbrack$ vs. \short: $\lbrack \mbox{short mode} \rbrack$ following line</p>	<p>example of a long text line: long mode vs. short: short mode following line</p>
--	--

shortmode (*key*) T_EX 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
<code>off</code>	disabled
<code>above</code>	above short equations (package default)
<code>belowone</code>	also below short single-line equations
<code>belowall</code>	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`.

<p>example of a long text line: $\lbrack\lbrack\short\rbrack \mbox{forced short} \rbrack$ and short: $\lbrack\lbrack\long\rbrack \mbox{forced long} \rbrack$ following line</p>	<p>example of a long text line: forced short and short: forced long following line</p>
---	--

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.

<p>$\hrule\begin{minipage}{\linewidth}$ $\lbrack \mbox{top} \rbrack$ some text\par $\lbrack \mbox{par} \rbrack$ $\lbrack \mbox{cont} \rbrack$ $\end{minipage}\hrule$</p>	<p>top some text par cont</p>
---	--

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
medskip (*key*)

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^AT_EX 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 `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 <code>par</code> 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 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 boolean switches `strut` and `struttag`.

`displaybreak` (*key*) Finally, the breaking of multi-line equations across pages can be controlled as follows: The setting `\displaybreak` taking values 0 (never) through 4 (permissive) controls the permissivity of page breaks within multi-line equations. The optional argument `displaybreak`

taking values 0 (do not) through 4 (enforce) suggests a break just *above* the equation environment. The command `\displaybreak` with values 0 through 4 suggests a break below the current line or below the equation environment.

2.7 Further Environments

The package supplies some additional environments:

`equationsbox` (*env.*) The package provides a boxed equation environment `equationsbox` which can be used within arbitrary math content. It works analogously to `equations` including optional arguments `margin` (*key*) and modifiers, but it offers a reduced range of functionality such as (evidently) no numbering `marginleft` (*key*) and `marginright` (*key*) (yet, the `lines` mode accepts multiple columns here). Additional arguments are given by `margin`, `marginleft`, `marginright` which specify additional margin space around the equations box.

```

\[\left\{
\begin{equationsbox}[margin=2em]
  x &= \cos\phi \\
\phi &= \arccos x
\end{equationsbox}
\right\}\]

```

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

```

\[\left[
\begin{equationsbox}~[shape=1]
  x = \cos\phi &
\phi = \arccos x \\
x = (z+z^{-1})/2 &
\phi = -i\log z
\end{equationsbox}
\right]\]

```

$$\left[\begin{array}{ll} x = \cos \phi & \phi = \arccos x \\ x = (z + z^{-1})/2 & \phi = -i \log z \end{array} \right]$$

`subequations` (*env.*) The environment `subequations` group 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:

`subeqtemplate={#1\alph{#2}}`

```

\eqnlineset
  {subeqtemplate={#1-\roman{#2}}}
\begin{subequations}
\left[ x = \cos\phi \right] \quad \text{and} \quad x = \cos \phi \quad (54-i)
\text{and} \quad \phi = \arccos x \quad (54-ii)
\left[ \phi = \arccos x \right]
\end{subequations}

```

`intertext` (*env.*) The environment `intertext` (or 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
\intertext[medskip]{and}
\phi &= \arccos x \>

```

$$\begin{array}{ll} x = \cos \phi & \\ \text{and} & \\ \phi = \arccos x & \end{array}$$

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 ^A T _E X/amsmath (layout and dimensions)
<code>defaults=eqnlines</code>	eqnlines 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
<code>ampproof</code>	equip optional argument parsing with protection for <code>&</code>
<code>crerror</code>	invoke an error when <code>'\'</code> is used in a single equation

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

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

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

3.3 Files and Installation

The package consists of the files:

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

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

- Run (pdf)L^AT_EX on `eqnlines.dtx` to compile the manual `eqnlines.pdf` (this file).
- Run L^AT_EX on `eqnlines.ins` to create the package `eqnlines.sty` and the developers version `eqnlines-dev.sty`. Copy the file `eqnlines.sty` to an appropriate directory of your L^AT_EX 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` will rename the math environments of `amsmath` with a prefix `ams...` and overwrite them with its own implementations. 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 adding a box and 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 v7.011 (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:

- documentation for all features
- complete code documentation
- switch to allow scanning of `\par` within body
- numbering scheme to place a number where there is the most available space

3.6 Revision History

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 field
- `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^AT_EX/`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 \@currrenvir}}
6 <dev>\def\eql@dev@leaveenv{\eql@dev{ leaving \@currrenvir}}
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^AT_EX 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 ϵ -T_EX 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^AT_EX defines `\dollar@begin` and `\dollar@end` to represent (and adjust) the beginning and end of bare T_EX 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^AT_EX `\@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^AT_EX’s original `\@ifnextchar` in `\kernel@ifnextchar` which skips over spaces as `\eql@ifnextchar@loose`. We replicate the `amsgn` version `\new@ifnextchar` that does not skip over spaces as `\eql@ifnextchar@loose`. The space before #1 allows to look-ahead for spaces as well:

```
17 \let\eql@ifnextchar@loose\kernel@ifnextchar
18 \long\def\eql@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 `\eq@ifstar@...` macros `\eq@ifstar@...` and `\eq@testopt@...` replicate the L^AT_EX counterparts `\eq@testopt@...` `\@ifstar` and `\@testopt`. The macros `\eq@ifnextgobble@...` work like `\@ifnextchar`, `\eq@teststaropt@...` 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}{%
41 \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
42 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{%
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}}}

```

`\eq@spbgroup` The second challenge is addressed by enclosing the look-ahead in spurious groups³ which `\eq@speggroup` protect against triggering ‘&’. The macros `\eq@spbgroup` and `\eq@speggroup` open and `\eq@srbgroup` close a spurious group. For some reason, the look-ahead mechanism requires further `\eq@sregroup` protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end. These adjustments are included in the macros `\eq@srbgroup` and `\ers@speggroup`:

```

50 \def\eq@spbgroup{\iffalse{\fi\ifnum0='}\fi}
51 \def\eq@speggroup{\ifnum0='\fi\iffalse}\fi}
52 \def\eq@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
53 \def\eq@sregroup{\let\@let@token\relax\ifnum0='\fi\iffalse}\fi}

```

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

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

```
54 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1{\eql@sregroup#2}}
55 \long\def\eql@ampprotecttwo#1#2#3{%
56   \eql@srbgroup#1{\eql@sregroup#2}{\eql@sregroup#3}}
```

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

```
57 \def\eql@ifnextchar@loose@ampsafe#1{%
58   \eql@ampprotecttwo{\eql@ifnextchar@loose#1}}
59 \def\eql@ifnextchar@tight@ampsafe#1{%
60   \eql@ampprotecttwo{\eql@ifnextchar@tight#1}}
61 \def\eql@ifstar@loose@ampsafe{\eql@ampprotecttwo\eql@ifstar@loose}
62 \def\eql@ifstar@tight@ampsafe{\eql@ampprotecttwo\eql@ifstar@tight}
63 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
64 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
65 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
66 \long\def\eql@teststaropt@tight@ampsafe{%
67   \eql@ampprotecttwo\eql@teststaropt@tight}}
```

`\eql@amproof` We may want to replace L^AT_EX’s definitions `\@ifnextchar`, `\@ifstar` and `\@testopt` to `\eql@amprevert` respect ‘&’ characters within aligned equations. This might make unrelated definitions with optional arguments and starred variants more robust in this context. The macro `\eql@amproof` overwrites the original definitions, and `\eql@amprevert` reverts the changes:

```
68 \let\eql@ifnextchar@org\@ifnextchar
69 \let\eql@ifstar@org\@ifstar
70 \let\eql@testopt@org\@testopt
71 \def\eql@amprevert{%
72   \let\@ifnextchar\eql@ifnextchar@org
73   \let\@testopt\eql@testopt@org
74   \let\@ifstar\eql@ifstar@org
75 }
76 \def\eql@amproof{%
77   \let\@ifnextchar\eql@ifnextchar@loose@ampsafe
78   \let\@testopt\eql@testopt@loose@ampsafe
79   \let\@ifstar\eql@ifstar@loose@ampsafe
80 }
```

B.5 Error Messages

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

```
\eql@warning
81 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
82 \def\eql@warning{\PackageWarning{eqnlines}}
```

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

```
\eql@error@mathmode
83 \def\eql@error@nomathmode#1{\eql@error{#1 allowed only in math mode}}
84 \def\eql@error@mathmode#1{\eql@error{#1 allowed only in paragraph mode}}
```

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

```
\eql@warn@label@multiple
\warn@label@multiple
\warn@tag@unused
\warn@tag@multiple
85 \def\eql@warn@label@unused{\eql@warning{Unused equation \string\label:
86   label ‘\eql@nextlabel’ will be lost}}
87 \def\eql@warn@label@multiple#1{\eql@warning{Multiple equation \string\label’s:
88   previous label ‘#1’ will be lost}}
```

```

89 \def\eql@warn@tag@unused{\eql@warning{Unused equation \string\tag:
90     tag declaration will be lost}}
91 \def\eql@warn@tag@multiple{\eql@warning{Multiple equation \string\tag's:
92     previous tag declaration will be lost}}

```

B.6 amsmath Integration

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

```

93 \def\eql@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
94 \def\eql@amsmath@before#1{%
95     \ifpackageloaded{amsmath}{\AddToHook{package/amsmath/before}{#1}}
96 \def\eql@amsmath@undefine#1{\eql@amsmath@before{\let#1\undefined}}
97 \def\eql@amsmath@let#1#2{\eql@amsmath@undefine#1\let#1#2}

```

B.7 PDF Tagging Support

`\eql@tagging@...` Proper PDF tagging⁴ support requires a L^AT_EX version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with L^AT_EX versions around 2020:

```

98 \let\eql@tagging@on\eql@false
99 \IfFormatAtLeastTF{2025-06-01}{%
100     \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}{}
101 \ifdefined\eql@tagging@on
102     \def\eql@tagging@mathsave{%
103         \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}
104     \def\eql@tagging@mathaddlast{%
105         \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
106     \def\eql@tagging@tagbegin{%
107         \UseTaggingSocket{math/display/tag/begin}}
108     \def\eql@tagging@tagend{%
109         \UseTaggingSocket{math/display/tag/end}}
110     \def\eql@tagging@tagsave{%
111         \UseTaggingSocket{math/luamml/mtable/tag/save}}
112     \def\eql@tagging@tagaddbox{%
113         \setbox\z@\copy\eql@tagbox%
114         \UseTaggingSocket{math/luamml/mtable/tag/set}}
115     \def\eql@tagging@tablesaveinner{%
116         \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
117     \def\eql@tagging@tableaddinner{%
118         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
119     \def\eql@tagging@tablesavealign{%
120         \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
121     \def\eql@tagging@tablesavealign{%
122         \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
123     \def\eql@tagging@alignleft{%
124         \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
125     \def\eql@tagging@aligncenter{%
126         \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}

```

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

```

127 \def\eql@tagging@alignright{%
128   \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:

```

129 \let\eql@single@doscan\eql@true
130 \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:

```

```

131 \ExplSyntaxOn
132 \def\eql@tagging@start{%
133   \bool_if:NF\l__math_collected_bool{%
134     \toks@\expandafter{\eql@tagging@opt}%
135     \edef\eql@tmp{\@currentenv}{[\the\toks@] \the\eql@scan@reg}}%
136     \expandafter\__math_process:nn\eql@tmp
137     \@kernel@math@registered@begin
138     \bool_set_true:N\l__math_collected_bool
139   }%
140 }
141 \def\eql@tagging@end{}
142 \ExplSyntaxOff
143 \else
144 \def\eql@tagging@mathsave{}
145 \def\eql@tagging@mathaddlast{}
146 \def\eql@tagging@tagbegin{}
147 \def\eql@tagging@tagend{}
148 \def\eql@tagging@tagsave{}
149 \def\eql@tagging@tagaddbox{}
150 \def\eql@tagging@tablesaveinner{}
151 \def\eql@tagging@tableaddinner{}
152 \def\eql@tagging@tablesavealign{}
153 \def\eql@tagging@tablesavealign{}
154 \def\eql@tagging@alignleft{}
155 \def\eql@tagging@aligncenter{}
156 \def\eql@tagging@alignright{}
157 \def\eql@tagging@start{}
158 \def\eql@tagging@end{}
159 \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:

```
160 \let\eql@tagsleft\eql@false
```

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

```
161 \let\eql@layoutleft\eql@false
```

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

`\eql@layoutleftmarginmin@` (*dimen*)

```
162 \newdimen\eql@layoutleftmargin@
```

```
163 \newdimen\eql@layoutleftmarginmin@
```

```
164 \newdimen\eql@layoutleftmarginmax@
```

```
165 \def\eql@layoutleftmargin@val{\leftmargini}
```

```
166 \eql@layoutleftmarginmin@\z@
```

```
167 \eql@layoutleftmarginmax@.5\maxdimen
```

`\eql@tagmargin@` (*dimen*) The intended margin width for tags in central alignment layout is stored in

`\eql@tagmargin@ratio@` (*dimen*) `\eql@tagmargin@` which is sourced by `\eql@tagmargin@val`. An undefined

`\eql@tagmargin@val` `\eql@tagmargin@val` will compute the margin width as the maximum width of tags

`\eql@tagmargin@threshold` (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

```
168 \newdimen\eql@tagmargin@
```

```
169 \let\eql@tagmargin@val\undefined
```

```
170 \newdimen\eql@tagmargin@ratio@
```

```
171 \eql@tagmargin@ratio@\p@
```

```
172 \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:

```
173 \newdimen\eql@indent@
```

```
174 \def\eql@indent@val{2em}
```

`\eql@paddingleft@` (*dimen*) The padding of an equation (column) is specified by `\eql@paddingleft@` and

`\eql@paddingright@` (*dimen*) `\eql@paddingright@`. These dimension registers are set to the macros

`\eql@paddingleft@val` and `\eql@paddingright@val`, respectively, when entering the equation environments:

```
175 \newdimen\eql@paddingleft@
```

```
176 \newdimen\eql@paddingright@
```

```
177 \def\eql@paddingleft@val{0pt}
```

```
178 \def\eql@paddingright@val{0pt}
```

`\eql@paddingmax` (*bool*) The boolean register `\eql@paddingmax` specifies whether the full line should be used for padding:

```
179 \let\eql@paddingmax\eql@false
```

`\eql@box@marginleft` and `\eql@box@marginright` The macros `\eql@box@marginleft` and `\eql@box@marginright` specify the margin surrounding equation boxes:

```
180 \def\eql@box@marginleft{\z@skip}
```

```
181 \def\eql@box@marginright{\z@skip}
```

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

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

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

```
183 \def\eql@spread@val{\jot}
184 \newdimen\eql@spread@
```

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

```
185 \newdimen\eql@tagfuzz@
186 \eql@tagfuzz@16sp\relax
```

`\eql@display@height` An equation will appear to the surrounding text with a fixed apparent height and depth specified by `\eql@display@height` and `\eql@display@depth`, respectively. By default it appears as a strut for equations:

```
187 \def\eql@display@height{\ht\eql@strutbox@}
188 \def\eql@display@depth{\dp\eql@strutbox@}
```

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

```
189 \def\eql@skip@mode@short{2}

190 \def\eql@skip@mode@cont@above{2}
191 \def\eql@skip@mode@cont@below{0}

192 \def\eql@skip@mode@par@above{3}
193 \def\eql@skip@mode@par@below{0}

194 \def\eql@skip@mode@top@above{4}
195 \def\eql@skip@mode@top@below{0}

196 \newcount\eql@skip@mode@leave@
197 \let\eql@skip@force@leave@\undefined
```

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

`\eql@skip@force@below`

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

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

```
198 \newcount\eql@skip@mode@above@
199 \newcount\eql@skip@mode@below@
200 \let\eql@skip@force@above@\undefined
201 \let\eql@skip@force@below@\undefined
202 \let\eql@skip@custom@above@\undefined
203 \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`:

```

204 \def\eql@skip@long@above{\abovedisplayskip}
205 \def\eql@skip@long@below{\belowdisplayskip}
206 \def\eql@skip@short@above{\abovedisplaysshortskip}
207 \def\eql@skip@short@below{\belowdisplaysshortskip}
208 \def\eql@skip@cont@above{\eql@skip@short@above}
209 \def\eql@skip@cont@below{\eql@skip@short@below}
210 % \TODO: parabove plus parskip?
211 \def\eql@skip@par@above{\eql@skip@long@above}
212 \def\eql@skip@par@below{\eql@skip@long@below}
213 \def\eql@skip@top@above{\eql@skip@long@above}
214 \def\eql@skip@top@below{\eql@skip@long@below}
215 \def\eql@skip@med@above{\abovedisplayskip/2}
216 \def\eql@skip@med@below{\belowdisplayskip/2}
217 \def\eql@skip@tag@above{\z@skip}
218 \def\eql@skip@tag@below{\z@skip}
219 \def\eql@skip@partag@above{\z@skip}
220 \def\eql@skip@partag@below{\z@skip}
221 \def\eql@skip@medtag@above{\z@skip}
222 \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:

```

223 \newdimen\eql@colsepmin@
224 \def\eql@colsepmin@val{1em}
225 \def\eql@colsepmax@val{.5\maxdimen}

```

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

```

226 \newdimen\eql@tagwidthmin@
227 \eql@tagwidthmin@\z@

```

`\eql@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eql@tagsepmin@`. T_EX's built-in value is half a quad⁵ in font number 2. As the tag is processed in text mode, we use 0.5em instead.

```

228 \newdimen\eql@tagsepmin@
229 \def\eql@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

`\eql@equations@sqr@opt` The macros `\eql@equations@sqr@opt` and `\eql@equations@ang@opt` store the default arguments for `\[...]` and `\<...>`, respectively:

```

230 \def\eql@equations@sqr@opt{equation,nonumber}
231 \def\eql@equations@ang@opt{align,nonumber}

```

Multi-Line Align Mode.

```

232 \let\eql@columns@margins\eql@true

```

⁵another half of a quad is left empty at the other end of the line.

C.2 Registers

TODO: describe

General.

`\eq@fieldbox@` (*box*) The box `\eq@fieldbox@` holds the present alignment component and `\eq@tagbox@` the tag for the present line. The corresponding dimensions `\eq@fieldwidth@` and `\eq@tagwidth@` (*dimen*) hold their widths:

```
233 \newbox\eq@fieldbox@
234 \newbox\eq@tagbox@
235 \newdimen\eq@fieldwidth@
236 \newdimen\eq@tagwidth@
```

`\eq@totalwidth@` (*dimen*)
`\eq@tagwidth@max@` (*dimen*)

```
237 \newdimen\eq@totalwidth@
238 \newdimen\eq@tagwidth@max@
```

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

```
239 \newdimen\eq@line@height@
240 \newdimen\eq@line@depth@
```

`\eq@line@width@` (*dimen*)
`\eq@line@avail@` (*dimen*)
`\eq@line@pos@` (*dimen*)

```
241 \newdimen\eq@line@width@
242 \newdimen\eq@line@avail@
243 \newdimen\eq@line@pos@
```

Rows and Columns.

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

```
244 \newcount\eq@row@
245 \newcount\eq@totalrows@
246 \newcount\eq@tagrows@
```

`\eq@column@`
`\eq@totalcolumns@`

```
247 \newcount\eq@column@
248 \newcount\eq@totalcolumns@
```

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

```
249 \newdimen\eq@colsep@
```

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

```
250 \newcount\eq@columns@inter@
```

Vertical Spacing Adjustments.

`\firstavail@` (*dimen*) The unused space on the first line of an alignment is stored in `\eql@display@firstavail@` for comparison against `\prel@display@firstavail@` 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:

```
251 \newdimen\eql@display@firstavail@
252 \def\eql@display@firstavail@set#1{%
253   \ifnum\eql@row@=\@one
254     \global\eql@display@firstavail@#1%
255   \fi
256 }
```

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

```
257 \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:

```
258 \ifdefined\measuring@true\else
259   \expandafter\newif\csname ifmeasuring@\endcsname
260 \fi
261 \AddToHook{package/hyperref/after}{
262   \ifdefined\Hy@ifnotmeasuring
263     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
264   \fi
265 }
```

`\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:

```
266 \ifdefined\@displaytrue\else
267   \expandafter\newif\csname if@display\endcsname
268   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
269 \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 \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

```
270 \let\eql@hook@blockbefore\@empty
271 \let\eql@hook@blockafter\@empty
272 \let\eql@hook@blockin\@empty
273 \let\eql@hook@blockout\@empty
```



```

274 \let\eq@hook@linein\@empty
275 \let\eq@hook@lineout\@empty
276 \let\eq@hook@colin\@empty
277 \let\eq@hook@colout\@empty
278 \let\eq@hook@eqin\@empty
279 \let\eq@hook@eqout\@empty
280 \let\eq@hook@innerleft\@empty
281 \let\eq@hook@innerright\@empty
282 \let\eq@hook@innerlead\@empty

```

D 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.

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

```

\eq@punct@line
\eq@punct@block
283 \let\eq@punct@col\@empty
284 \let\eq@punct@line\relax
285 \let\eq@punct@block\relax

```

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

```

286 \let\eq@punct@sep\relax

```

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

```

\eqnpunct
287 \def\eqnpunctcol{\eq@ifstar@tight\eq@punct@col@setrelax\eq@punct@col@set}
288 \def\eq@punct@col@set#1{\def\eq@punct@col{#1}\ignorespaces}
289 \def\eq@punct@col@setrelax{\let\eq@punct@col\@empty\ignorespaces}
290 \def\eqnpunctline{\eq@ifstar@tight\eq@punct@line@setrelax\eq@punct@line@set}
291 \def\eq@punct@line@set#1{\def\eq@punct@line{#1}\ignorespaces}
292 \def\eq@punct@line@setrelax{\let\eq@punct@line\relax\ignorespaces}
293 \def\eqnpunctmain{\eq@ifstar@tight\eq@punct@main@setrelax\eq@punct@main@set}
294 \def\eq@punct@main@set#1{\def\eq@punct@main{#1}\ignorespaces}
295 \def\eq@punct@main@setrelax{\let\eq@punct@main\relax\ignorespaces}
296 \def\eqnpunct{\eq@ifstar@tight\eq@punct@next@setrelax\eq@punct@next@set}
297 \def\eq@punct@next@set#1{%
298   \ifmmode
299     \def\eq@punct@col{#1}%
300     \def\eq@punct@line{#1}%
301     \def\eq@punct@block{#1}%
302   \else
303     \eqnadopt{punct={#1}}%
304   \fi
305   \ignorespaces}
306 \def\eq@punct@next@setrelax{%
307   \ifmmode
308     \let\eq@punct@block\relax
309   \else
310     \eqnadopt{punct*}%
311   \fi
312   \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:

```

313 \def\eql@punct@apply@col{%
314   \ifx\eql@punct@col\@empty\else
315     \eql@punct@sep
316     \eql@punct@col
317     \let\eql@punct@col\@empty
318   \fi
319 }
```

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`

```

320 \def\eql@punct@apply@line{%
321   \ifx\eql@punct@line\relax
322 % \TODO hand down immediately?
323   \else
324     \ifx\eql@punct@line\@empty\else
325       \eql@punct@sep
326       \eql@punct@line
327     \fi
328     \let\eql@punct@line\relax
329     \let\eql@punct@col\@empty
330   \fi
331 }
```

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

```

332 \def\eql@punct@apply@block{%
333   \ifx\eql@punct@block\relax
334 % \TODO hand down immediately?
335   \else
336     \ifx\eql@punct@block\@empty\else
337       \eql@punct@sep
338       \eql@punct@block
339     \fi
340     \let\eql@punct@block\relax
341     \let\eql@punct@line\relax
342     \let\eql@punct@col\@empty
343   \fi
344 }
345 \let\eqnpunctapply\eql@punct@apply@block
```

E 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:

```

346 \def\eql@class@innerright@sel@{%
```

```

347 \ifdim\eql@fieldwidth@=\z@
348   \eql@class@innerlead
349 \else
350   \eql@class@innerright
351 \fi
352 }

```

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

```

353 \def\eql@class@innerleft@set#1{%
354   \def\eql@class@innerleft{#1}%
355 }
356 \def\eql@class@innerright@set#1{%
357   \def\eql@class@innerright{#1}%
358   \let\eql@class@innerright@sel\eql@class@innerright
359 }
360 \def\eql@class@innerlead@set#1{%
361   \def\eql@class@innerlead{#1}%
362   \let\eql@class@innerright@sel\eql@class@innerright@sel@
363 }

```

`\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`):

```

364 \def\eql@class@ampeq{%
365   \eql@class@innerleft@set{}%
366   \eql@class@innerright@set{{}}%
367 }
368 \def\eql@class@eqamp{%
369   \eql@class@innerleft@set{\mathrel{}}%
370   \eql@class@innerright@set{\mathrel{}}%
371   \eql@class@innerlead@set{{}}%
372 }
373 \eql@class@ampeq

```

F Equation Numbering

TODO: describe

F.1 Tag Formatting

TODO: describe

```

374 \def\eql@tag@setbox#1{%
375   \def\eql@tag@box##1{#1}%
376 }
377 \def\eql@tag@setbox#1{%
378   \def\eql@tag@box##1{\hbox{\m@th\normalfont#1}}%
379 }

```

TODO: describe

```

380 \def\eql@tag@setform#1{%
381   \def\eql@tag@form##1{#1}%
382 }

```

```

383 \def\eq@tag@setform#1#2#3{%
384   \def\eq@tag@form##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
385 }

386 \eq@tag@setbox{#1}
387 \eq@tag@setform{#1}
388 \def\eq@tag@boxedform#1{\eq@tag@box{\eq@tag@form{#1}}}

389 \DeclareRobustCommand{\tagform}{\eq@tag@form}
390 \DeclareRobustCommand{\tagbox}{\eq@tag@box}
391 \DeclareRobustCommand{\tagboxed}{\eq@tag@boxedform}

```

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

```

392 \DeclareRobustCommand{\eq@eqref}[1]{\textup{\eq@tag@boxedform{\ref{#1}}}}

```

Raise Tags.

`setag@amount@` (*dimen*)

```

393 \newdimen\eq@raisetag@amount@

```

`\raisetag`

```

394 \def\eq@raisetag@default{%
395   \eq@warning{string\raisetag\space not allowed here}
396   \@gobble
397 }

```

TODO: describe

```

398 \eq@amsmath@let\raisetag\eq@raisetag@default

```

TODO: maybe introduce a star form to enforce raise?

```

399 \def\eq@raisetag#1{\global\eq@raisetag@amount@\glueexpr#1\relax}
400 \let\eq@raisetag@gobble\@gobble

```

F.2 Showkeys Integration

TODO: describe

```

401 \let\eq@SK@loaded\eq@false
402 \let\eq@SK@label\@gobble
403 \let\eq@SK@clearlabel\@empty
404 \let\eq@SK@setlabel\@gobble
405 \let\eq@SK@printlabel@right\@empty
406 \let\eq@SK@printlabel@left\@empty
407 \let\eq@SK@printlabel@line\@empty
408 \def\eq@label@clean{\eq@label@org}
409 \AddToHook{package/showkeys/after}{
410   \let\eq@SK@loaded\eq@true
411   \def\eq@SK@label#1{\SK@SK@label#1}
412   \def\eq@SK@clearlabel{\let\eq@SK@lab\relax}
413   \eq@SK@clearlabel
414   \def\eq@SK@@label#1>#2\SK@{%
415     \def\eq@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%

```

```

416 }
417 \def\eqL@SK@setLabel#1{\SK@\eqL@SK@label#1}
418 \def\eqL@SK@printlabel@right{%
419   \ifx\eqL@SK@lab\relax\else
420     \rlap{\kern\marginparsep\eqL@SK@lab}%
421     \eqL@SK@clearlabel
422   \fi
423 }
424 \def\eqL@SK@printlabel@left{%
425   \ifx\eqL@SK@lab\relax\else
426     \llap{\eqL@SK@lab\kern\marginparsep}%
427     \eqL@SK@clearlabel
428   \fi
429 }
430 \def\eqL@SK@printlabel@line{%
431   \ifx\eqL@SK@lab\relax\else
432     \dimen@\prevdepth
433     \nointerlineskip
434     \ifdefined\eqL@tagsleft
435       \llap{%
436         \eqL@SK@lab
437         \kern\marginparsep
438       }%
439       \eqL@SK@clearlabel
440     \else
441       \rlap{%
442         \dimen@\displaywidth
443         \advance\dimen@\marginparsep
444         \kern\dimen@
445         \eqL@SK@lab
446       }%
447     \fi
448     \eqL@SK@clearlabel
449     \prevdepth\dimen@
450   \fi
451 }
452 \let\eqL@label@org\label
453 \def\eqL@label@clean{\let\SK@\@gobbletwo\eqL@label@org}
454 }

```

F.3 Labels

TODO: describe

```

455 % \TODO implement (via label[] or labelname similar to label/tag)
456 \let\eqL@nextlabel\@undefined
457 \def\eqL@labelname@default{[equation]}

```

\eqL@label@org

```

458 \let\eqL@label@org\label

```

F.4 Tags

TODO: describe

```

459 \let\eqL@nexttag\@undefined

```

`\eql@tag@default`

```
460 \def\eql@tag@default{%
461   \eql@error{\string\tag\space not allowed here}{}\eql@tag@gobble}
462 \let\tag\eql@tag@default
```

`\eql@tag@gobble` **TODO:** ifnextchar, gobbletwo?

```
463 \def\eql@tag@gobble@[#1]#2{}
464 \def\eql@tag@gobble{%
465   \eql@ampprotecttwo\eql@teststaropt@tight\eql@tag@gobble@\eql@tag@gobble@{}}
```

`\eql@nexttag` **TODO:** can amsmath handle also counter refstepcounter in tags?

`\eql@tag@makenext` hyperref anchors

`\eql@tag@makenext@@`

`\eql@tag@makenext@@@`

```
466 \let\eql@Hy@anchor@\gobble
467 \AddToHook{package/hyperref/after}{
468   \def\eql@Hy@anchor#1{%
469     \Hy@raisedlink{\hyper@anchor{#1}}%
470   }%
471 }

472 \def\eql@tag@makenext{%
473   \eql@ampprotecttwo\eql@teststaropt@tight
474   \eql@tag@makenext@star\eql@tag@makenext@\eql@tag@text
475 }
```

TODO: not sure about `\protected@edef\eql@tag@text` was `\def` only

```
476 \def\eql@tag@makenext@star[#1]#2{%
477   \global\def\eql@nexttag{%
478     \let\eql@tag@tool\@firstofone
479     \protected@edef\eql@tag@text{#2}%
480     \protected@edef\eql@tag@label{#1}%
481   }%
482 }
483 \def\eql@tag@makenext@[#1]#2{%
484   \global\def\eql@nexttag{%
485     \let\eql@tag@tool\eql@tag@form
486     \protected@edef\eql@tag@text{#2}%
487     \protected@edef\eql@tag@label{#1}%
488     \protected@edef\eql@tag@label{\p@equation\eql@tag@label}%
489   }%
490 }
```

F.5 Anchors

TODO: describe

`\eql@refcount@` (*counter*)

```
491 \newcount\eql@numbering@refcount@
492 \eql@numbering@refcount@\z@
493 \def\eql@numbering@ref{equation.eql-\the\eql@numbering@refcount@}
494 \def\eql@numbering@refstep{\global\advance\eql@numbering@refcount@\@ne}
```

TODO: describe

```
495 \def\eql@numbering@makeblockanchor{%
```

```

496 \eql@numbering@refstep
497 \global\edef\eql@label@currentHref{\eql@numbering@ref}%
498 \eql@Hy@anchor\eql@label@currentHref
499 \global\edef\eql@label@thepage{\thepage}%
500 }
501 \def\eql@numbering@setblockanchor{%
502 \let\thepage\eql@label@thepage
503 \let\@currentHref\eql@label@currentHref
504 }

```

F.6 Tag Composition

TODO: describe

```

\eql@compose@anchor
  \eql@compose@tag
  \eql@compose@label
505 \def\eql@compose@anchor{%
506 \ifdefined\eql@nexttag
507 \eql@nexttag
508 \def\@currentcounter{equation}%
509 \let\@currentlabel\eql@tag@label
510 \eql@numbering@refstep
511 \edef\@currentHref{\eql@numbering@ref}%
512 \eql@Hy@anchor\@currentHref
513 \global\let\eql@nexttag\@undefined
514 \else
515 \refstepcounter{equation}%
516 \let\eql@tag@tool\eql@tag@form
517 \edef\eql@tag@text{\theequation}%
518 \fi
519 }

520 \def\eql@compose@label{%
521 \ifmeasuring@\else
522 \eql@SK@clearlabel
523 \ifdefined\eql@nextlabel
524 \ifnum
525 \ifnum\eql@numbering@target@<\z@
526 \eql@row@
527 \else
528 \eql@numbering@target@
529 \fi=\eql@row@
530 \eql@compose@label@
531 \fi
532 \fi
533 \fi
534 }

```

TODO: describe

```

535 \def\eql@compose@label@{%
536 \eql@SK@setlabel\eql@nextlabel
537 \begingroup
538 \ifnum\eql@numbering@target@=\eql@row@
539 \eql@numbering@setblockanchor
540 \fi
541 \let\@currentlabelname\eql@labelname@default
542 \expandafter\eql@label@clean\expandafter{\eql@nextlabel}%

```

```

543 \global\let\eql@nextlabel\@undefined
544 \endgroup
545 }

```

TODO: describe

```

546 \def\eql@compose@tag{%
547 \eql@tagging@tagbegin
548 \eql@tag@box{%
549 \eql@tag@tool\eql@tag@text
550 \eql@tagging@tagsave
551 }%
552 \eql@tagging@tagend
553 }

```

TODO: describe

```

554 \def\eql@compose@print{%
555 \eql@compose@anchor
556 \eql@compose@label
557 \ifdefined\eql@tagsleft
558 \eql@SK@printlabel@left
559 \eql@compose@tag
560 \else
561 \eql@compose@tag
562 \eql@SK@printlabel@right
563 \fi
564 }

```

TODO: describe

```

565 \def\eql@compose@measure{%
566 \ifdefined\eql@nexttag
567 \eql@nexttag
568 \eql@tag@box{\eql@tag@tool\eql@tag@text}%
569 \else
570 \stepcounter{equation}%
571 \eql@tag@box{\eql@tag@form\theequation}%
572 \fi
573 \ifnum\eql@numbering@target@<\z@
574 \global\let\eql@nextlabel\@undefined
575 \global\let\eql@nexttag\@undefined
576 \fi
577 }

```

F.7 Tagbox Methods

TODO: describe

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

```

578 \def\eql@tagbox@make#1{%
579 \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
580 \eql@tagwidth@\wd\eql@tagbox@
581 \ifdim\eql@tagwidth@<\eql@tagwidthmin@
582 \eql@tagwidth@\eql@tagwidthmin@
583 \fi
584 \advance\eql@tagwidth@\eql@tagsepmin@
585 }

```


TODO: describe

```
586 \def\eq@tagbox@print@tagsright{%
587   \kern-\wd\eq@tagbox@
588   \box\eq@tagbox@
589 }
```

TODO: describe

```
590 \def\eq@tagbox@print@tagsleft{%
591   \wd\eq@tagbox@\z@
592   \box\eq@tagbox@
593 }
```

TODO: describe

```
594 \def\eq@tagbox@print@tagsright@raise{%
595   \ifnum\eq@row@=\eq@totalrows@
596     \global\advance\eq@raisetag@firstlast@\tw@
597     \fi
598   \llap{\vtop{%
599     \vskip-\eq@raisetag@amount@
600     \normalbaselines
601     \setbox\@ne\hbox{}}%
602     \dp\@ne\eq@line@depth@
603     \box\@ne
604     \box\eq@tagbox@
605   }}%
606 }
607 \def\eq@tagbox@print@tagsleft@raise{%
608   \ifnum\eq@row@=\@ne
609     \global\advance\eq@raisetag@firstlast@\@ne
610     \fi
611   \rlap{\vbox{%
612     \normalbaselines
613     \box\eq@tagbox@
614     \vbox to\eq@line@height@{}%
615     \vskip\eq@raisetag@amount@
616   }}%
617 }
```

TODO: describe

```
618 \def\eq@numbering@printsubeqlabel{%
619   \ifdefined\eq@parentlabel
620     \eq@numbering@makeblockanchor
621     \eq@SK@setlabel\eq@parentlabel
622     \begingroup
623       \def\@currentcounter{equation}%
624       \eq@numbering@setblockanchor
625       \let\@currentlabelname\eq@labelname@default
626       \protected@edef\@currentlabel{\p@equation\theparentequation}%
627       \expandafter\eq@label@clean\expandafter{\eq@parentlabel}%
628     \endgroup
629     \eq@SK@printlabel@line
630   \fi
631 }
```

F.8 Numbering Schemes

TODO: describe

```
632 \def\eq@numbering@tab@first{first}
633 \def\eq@numbering@tab@last{last}
634 \def\eq@numbering@tab@middle{middle}
635 \def\eq@numbering@tab@here{here}
636 \def\eq@numbering@tab@in{in}
637 \def\eq@numbering@tab@out{out}
638 \def\eq@numbering@tab@sub{sub}
639 \def\eq@numbering@tab@all{all}
640 \def\eq@numbering@tab@none{none}
```

TODO: describe

```
641 \let\eq@numbering@tab@f\eq@numbering@tab@first
642 \let\eq@numbering@tab@l\eq@numbering@tab@last
643 \let\eq@numbering@tab@m\eq@numbering@tab@middle
644 \let\eq@numbering@tab@mid\eq@numbering@tab@middle
645 \let\eq@numbering@tab@o\eq@numbering@tab@out
646 \let\eq@numbering@tab@outside\eq@numbering@tab@out
647 \let\eq@numbering@tab@i\eq@numbering@tab@in
648 \let\eq@numbering@tab@inside\eq@numbering@tab@in
649 \let\eq@numbering@tab@within\eq@numbering@tab@in
650 \let\eq@numbering@tab@h\eq@numbering@tab@here
651 \let\eq@numbering@tab@s\eq@numbering@tab@sub
652 \let\eq@numbering@tab@subeq\eq@numbering@tab@sub
653 \let\eq@numbering@tab@subequation\eq@numbering@tab@sub
654 \let\eq@numbering@tab@subequations\eq@numbering@tab@sub
655 \let\eq@numbering@tab@a\eq@numbering@tab@all
656 \let\eq@numbering@tab@n\eq@numbering@tab@none
657 \expandafter\let\csname eq@numbering@tab@!\endcsname\eq@numbering@tab@all
658 \expandafter\let\csname eq@numbering@tab@*\endcsname\eq@numbering@tab@none
659 \expandafter\let\csname eq@numbering@tab@1\endcsname\eq@numbering@tab@first

660 \let\eq@numbering@mode\eq@numbering@tab@all

661 \def\eq@numbering@set#1{%
662   \ifcsname eq@numbering@tab@#\endcsname
663     \expandafter\let\expandafter\eq@numbering@mode
664       \csname eq@numbering@tab@#\endcsname
665     \ifx\eq@numbering@mode\eq@numbering@tab@none
666       \let\eq@numbering@mode\eq@numbering@tab@all
667       \let\eq@numbering@active\eq@false
668     \fi
669   \else
670     \eq@error{numbering mode ‘#1’ unknown: setting to ‘all’}%
671     \let\eq@numbering@mode\eq@numbering@tab@all
672   \fi
673 }
```

ing@target@ (*counter*)

```
674 \let\eq@numbering@active\eq@true
675 \newcount\eq@numbering@target@

676 \def\eq@numbering@mode@all{%
677   \eq@numbering@target@\m@ne}
```

```

678 \def\eql@numbering@mode@sub{%
679   \eql@numbering@target@\m@ne
680   \let\eql@numbering@subeq@use\eql@true}
681 \def\eql@numbering@mode@first{%
682   \eql@numbering@target@\@ne}
683 \def\eql@numbering@mode@last{%
684   \eql@numbering@target@\@MM}
685 \def\eql@numbering@mode@here{%
686   \eql@numbering@target@\z@}

```

TODO: describe

```

687 \def\eql@numbering@mode@in{%
688   \ifdefined\eql@tagsleft
689     \eql@numbering@mode@last
690   \else
691     \eql@numbering@mode@first
692   \fi
693 }

```

TODO: describe

```

694 \def\eql@numbering@mode@out{%
695   \ifdefined\eql@tagsleft
696     \eql@numbering@mode@first
697   \else
698     \eql@numbering@mode@last
699   \fi
700 }

```

TODO: describe

```

701 \def\eql@numbering@mode@middle{%
702   \eql@numbering@target@\z@
703   \let\eql@numbering@eval@target\eql@numbering@eval@middle}
704 \def\eql@numbering@eval@middle{%
705   \ifnum\eql@numbering@target@=\z@
706     \count@\eql@row@
707     \advance\count@\@ne
708     \divide\count@\tw@
709     \global\eql@numbering@target@\count@
710   \fi
711 }

```

TODO: describe

```

712 \def\eql@numbering@eval@mode{%
713   \let\eql@numbering@eval@target\undefined
714   \let\eql@numbering@subeq@use\eql@false
715   \csname eql@numbering@mode@\eql@numbering@mode\endcsname
716   \ifdefined\eql@numbering@active
717     \let\eql@numbering@eqnswinit\@eqnswtrue
718   \else
719     \let\eql@numbering@eqnswinit\@eqnswfalse
720   \fi
721   \let\eql@numbering@active\eql@false
722 }

```

TODO: reconsider operation

\numberhere

```

723 \def\numberhere{%
724   \ifnum\eq@numbering@target@<\z@
725     \donumber
726   \else
727     \ifmeasuring@
728       \global\eq@numbering@target@\eq@row@
729     \fi
730   \fi
731 }

```

TODO: describe

`\numbernext`

```

732 \def\numbernext{%
733   \ifnum\eq@numbering@target@<\z@
734     \nonumber
735   \else
736     \ifmeasuring@
737       \ifnum\eq@numbering@target@=\eq@row@
738         \global\advance\eq@numbering@target@\@ne
739       \fi
740     \fi
741   \fi
742 }

```

F.9 Numbering Framework

TODO: describe

```

743 \let\eq@numbering@autolabel\eq@false
744 \let\eq@numbering@autotag\eq@true
745 \let\eq@numbering@blocklabel\@undefined
746 \let\eq@numbering@blocktag\@undefined
747 \let\eq@numbering@warn\eq@true

748 \def\eq@numbering@autoenable{%
749   \global\@eqnswtrue
750   \ifx\eq@numbering@mode\eq@numbering@tab@here
751     \ifnum\eq@numbering@target@=\z@
752       \numberhere
753     \fi
754   \fi
755 }

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

```

TODO: describe

```

760 \def\donumber{%
761   \if@eqnsw\else
762     \global\@eqnswtrue
763     \ifx\print@eqn\@empty
764       \global\let\print@eqn\eq@print@eqnum@default
765     \fi
766     \ifx\incr@eqn\@empty

```

```

767     \global\let\incr@eqn\eql@incr@eqnum@default
768     \fi
769 \fi
770 }

```

TODO: describe

```

771 \def\eql@label{%
772   \ifdefined\eql@numbering@autolabel
773     \eql@numbering@autoenable
774   \fi
775   \ifdefined\eql@numbering@warn
776     \ifdefined\eql@nextlabel
777       \eql@warn@label@multiple\eql@nextlabel
778     \fi
779   \fi
780 \global\edef\eql@nextlabel
781 }

```

TODO: describe

```

782 \let\eql@label@gobble\@gobble

```

TODO: describe

```

783 \def\eql@tag{%
784   \ifdefined\eql@numbering@autotag
785     \eql@numbering@autoenable
786   \fi
787   \ifdefined\eql@numbering@warn
788     \ifdefined\eql@nexttag
789       \eql@warn@tag@multiple
790     \fi
791   \fi
792   \eql@tag@makenext
793 }

```

TODO: describe

```

794 \def\eql@blocklabel@set#1{%
795   \ifdefined\eql@blocklabel
796     \eql@warn@label@multiple\eql@blocklabel
797   \fi
798   \edef\eql@blocklabel{#1}%
799 }

```

TODO: describe

```

800 \def\eql@blocktag@set#1{%
801   \ifdefined\eql@blocktag
802     \eql@warn@tag@multiple
803   \fi
804   \def\eql@blocktag{#1}%
805 }

```

TODO: describe

```

806 \def\eql@blocktag@setstar#1{%
807   \ifdefined\eql@blocktag
808     \eql@warn@tag@multiple
809   \fi
810   \def\eql@blocktag{*#1}%
811 }

```

Single Line. **TODO:** describe

```
812 \def\eq@numbering@single@init{%
813   \let\eq@numbering@warn\eq@true
814   \let\label\eq@label
815   \let\tag\eq@tag
816   \let\raisetag\eq@raisetag
817   \eq@numbering@target@{m@ne
818   \let\eq@nextlabel\eq@blocklabel
819   \ifdefined\eq@blocktag
820     \expandafter\eq@tag@makenext\eq@blocktag
821   \else
822     \let\eq@nexttag@undefined
823   \fi
824   \eq@numbering@eqnswinit
825   \ifdefined\eq@numbering@autolabel
826     \ifdefined\eq@nextlabel
827       \eqnswtrue
828     \fi
829   \fi
830   \ifdefined\eq@numbering@autotag
831     \ifdefined\eq@nexttag
832       \eqnswtrue
833     \fi
834   \fi
835   \global\eq@raisetag@amount@{z@
836 }
```

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

```
837 \def\eq@numbering@measure@init{%
838   \let\eq@numbering@warn\eq@true
839   \let\label\eq@label
840   \let\tag\eq@tag
841   \let\raisetag\eq@raisetag
842   \global\let\eq@nextlabel\eq@blocklabel
843   \ifdefined\eq@blocktag
844     \expandafter\eq@tag@makenext\eq@blocktag
845   \else
846     \global\let\eq@nexttag@undefined
847   \fi
848   \ifnum\eq@numbering@target@<{z@\else
849     \eq@numbering@eqnswinit
850     \ifdefined\eq@numbering@autolabel
851       \ifdefined\eq@nextlabel
852         \eqnswtrue
853       \fi
854     \fi
855   \fi
856 }
```

TODO: describe

```
857 \def\eq@numbering@measure@line@begin{%
858   \ifnum\eq@numbering@target@<{z@
859     \global\eq@numbering@eqnswinit
860   \fi
861 }
```

TODO: describe

```
862 \def\eq@numbering@measure@eval{%
863   \ifdefined\eq@numbering@eval@target
864     \eq@numbering@eval@target
865   \fi
866   \ifnum\eq@numbering@target@>\eq@row@
867     \global\eq@numbering@target@\eq@row@
868   \fi
869   \ifnum\eq@numbering@target@>\z@
870     \if@eqnsw\else
871       \global\eq@numbering@target@\z@
872     \fi
873   \fi
874   \ifnum\eq@numbering@target@<\@ne
875     \ifdefined\eq@nextlabel
876       \eq@warn@label@unused
877       \global\let\eq@nextlabel\@undefined
878     \fi
879     \ifdefined\eq@nexttag
880       \eq@warn@tag@unused
881       \global\let\eq@nexttag\@undefined
882     \fi
883   \fi
884 }
```

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

```
885 \def\eq@numbering@print@init{%
886   \ifnum\eq@numbering@target@<\z@
887     \let\eq@numbering@warn\eq@false
888     \let\label\eq@label
889     \let\tag\eq@tag
890     \let\raisetag\eq@raisetag
891     \let\eq@nextlabel\eq@blocklabel
892     \ifdefined\eq@blocktag
893       \expandafter\eq@tag@makenext\eq@blocktag
894     \else
895       \let\eq@nexttag\@undefined
896     \fi
897   \else
898     \let\label\eq@label@gobble
899     \let\tag\eq@tag@gobble
900     \let\raisetag\eq@raisetag@gobble
901   \fi
902 }
```

TODO: describe

```
903 \def\eq@numbering@print@block@begin{%
904   \ifnum\eq@numbering@target@>\z@
905     \eq@numbering@makeblockanchor
906   \fi
907   \ifdefined\eq@numbering@subeq@use
908     \eq@numbering@printsubeqlabel
909   \fi
910 }
```

TODO: describe

```

911 \def\eq@numbering@print@line@begin{%
912   \ifnum\eq@numbering@target@<\z@
913     \global\eq@numbering@eqnswinit
914     \global\eq@raisetag@amount@\z@
915   \fi
916 }

```

TODO: describe

```

917 \def\eq@numbering@print@line@eval{%
918   \ifnum\eq@numbering@target@<\z@\else
919     \ifnum\eq@numbering@target@=\eq@row@
920       \global\@eqnswtrue
921     \else
922       \global\@eqnswfalse
923     \fi
924   \fi
925 }

```

G Subequation Numbering

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

G.1 Definitions

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

```

926 \eq@amsmath@undefine\c@parentequation
927 \eq@amsmath@undefine\theparentequation
928 \ifdefined\c@parentequation\else
929 \newcounter{parentequation}
930 \fi

```

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

```

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

```

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

```

932 \let\eq@subequations@active\eq@false

```

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

```

933 \def\eq@subequations@init{%
934   \edef\eq@subequations@restorecounter{%
935     \global\c@equation\the\c@equation\relax}%

```



```

936 \ifdefined\eql@blocktag
937   \expandafter\eql@tag@makenext\eql@blocktag
938   \eql@nexttag
939   \eql@numbering@refstep
940   \protected@edef\theHparentequation{\eql@numbering@ref}%
941   \protected@edef\theparentequation{\eql@tag@text}%
942 \else
943   \advance\c@equation\@ne
944   \protected@edef\theparentequation{\theequation}%
945   \ifdefined\theHequation
946     \protected@edef\theHparentequation{\theHequation}%
947   \fi
948 \fi
949 \global\c@parentequation\c@equation
950 \global\c@equation\z@
951 \let\theequation\eql@subequations@template
952 \def\theHequation{\theHparentequation.\arabic{equation}}%
953 }

```

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

```

954 \def\eql@subequations@close{%
955   \ifnum\c@equation=\z@
956     \eql@subequations@restorecounter
957   \else
958     \global\c@equation\c@parentequation
959   \fi
960 }

```

G.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@printsubequlabel`

```

961 \def\eql@subequations@start{%
962   \let\eql@blocktag\@undefined
963   \let\eql@blocklabel\@undefined
964   \eql@nextopt@process{subequations}%
965   \eql@subequations@init
966   \eql@numbering@refstep
967   \edef\eql@subequations@currentHref{\eql@numbering@ref}%
968   \eql@Hy@anchor\eql@subequations@currentHref
969   \edef\eql@subequations@thepage{\thepage}%
970   \def\@currentcounter{equation}%
971   \let\@currentHref\eql@subequations@currentHref
972   \protected@edef\@currentlabel{\p@equation\theparentequation}%
973   \let\@currentlabelname\eql@labelname@default
974   \let\eql@subequations@active\eql@true
975   \ifdefined\eql@blocklabel
976     \eql@SK@label\eql@blocklabel
977   \fi
978   \ignorespaces
979 }

```

`eql@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:

```

980 \def\eql@subequations@end{%
981   \ifnum\c@equation>\z@
982     \ifdefined\eql@blocklabel
983       \begingroup
984         \def\@currentcounter{equation}%
985         \let\thepage\eql@subequations@thepage
986         \let\@currentHref\eql@subequations@currentHref
987 % \TODO how about tag* ?! also within equations!
988         \protected@edef\@currentlabel{\p@equation\theparentequation}%
989         \let\@currentlabelname\eql@labelname@default
990         \expandafter\eql@label@clean\expandafter{\eql@blocklabel}%
991       \endgroup
992     \fi
993   \fi
994   \eql@subequations@close
995   \ignorespacesafterend
996 }

```

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

```

997 \newenvironment{eql@subequations}{%
998 (dev)\eql@dev@enterenv
999   \eql@subequations@testall\eql@subequations@start%
1000 }{%
1001   \eql@subequations@end
1002 (dev)\eql@dev@leaveenv
1003 }

```

TODO: describe

```

1004 \def\eql@subequations@testall{\eql@subequations@testopt}
1005 \def\eql@subequations@testopt#1{%
1006   \eql@ifnextchar@tight[%]
1007     {\eql@subequations@adopt{\eql@subequations@testat{#1}}}%
1008     {\eql@subequations@testat{#1}}
1009 \def\eql@subequations@adopt#1[#2]{\eqnadopt{#2}#1}
1010 \def\eql@subequations@testat#1{%
1011   \eql@ifat@tight%
1012     {\eql@subequations@addlabel{#1}}%
1013     {#1}}
1014 \def\eql@subequations@addlabel#1#2{\eqnadopt{label={#2}}#1}

```

G.3 Subequation Scheme

TODO: describe

```

1015 \def\eql@numbering@subeq@init{%
1016   \let\eql@save@theequation\theequation
1017   \let\eql@save@theHequation\theHequation
1018   \eql@subequations@init
1019   \let\eql@parentlabel\eql@blocklabel
1020   \let\eql@parenttag\eql@blocktag
1021   \let\eql@blocklabel\@undefined
1022   \let\eql@blocktag\@undefined
1023 }

```

TODO: describe

```
1024 \def\eq@numbering@subeq@test{%
1025   \ifnum\c@equation<\tw@
1026     \let\eq@numbering@subeq@use\@ne
1027   \fi
1028 }
```

TODO: describe

```
1029 \def\eq@numbering@subeq@revert{%
1030   \let\eq@blocklabel\eq@parentlabel
1031   \let\eq@blocktag\eq@parenttag
1032   \let\eq@numbering@subeq@use\eq@false
1033   \let\theequation\eq@save@theequation
1034   \let\theHequation\eq@save@theHequation
1035   \eq@subequations@restorecounter
1036 }
```

TODO: describe

```
1037 % \TODO note must not use setcounter here (when calc is loaded)
1038 \def\eq@numbering@subeq@close{%
1039   \eq@subequations@close
1040 }
```

H Display Equations Support

TODO: describe

H.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1041 \interdisplaylinepenalty\@M
```

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

```
1042 \def\eq@getdsp@pen#1{%
1043   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1044 }
```

TODO: allow a displaybreak before equations

```
1045 \DeclareRobustCommand{\eq@displaybreak@default}[1][4]{%
1046   \eq@warning{Invalid use of \string\displaybreak}{}}
1047 \eq@amsmath@after{\let\eq@displaybreak@default\displaybreak}
1048 \eq@amsmath@let\displaybreak\eq@displaybreak@default

1049 \newcount\eq@displaybreak@pen@
1050 \newcount\eq@displaybreak@prepen@

1051 \protected\def\eq@displaybreak@print{%
1052   \eq@ampprotect\eq@testopt@tight\eq@displaybreak@print@{4}%
1053 }
```

TODO: describe

```
1054 \def\eqldisplaybreak@print@#1{%
1055   \ifnum#1<\z@
1056     \global\eqldisplaybreak@pen@\@MM
1057   \else
1058     \global\eqldisplaybreak@pen@-\@getpen{#1}\relax
1059   \fi
1060 }
```

TODO: describe

```
1061 \def\eqldisplaybreak@pre#1{%
1062   \ifnum#1<\z@
1063     \global\eqldisplaybreak@prepen@\@MM
1064   \else
1065     \global\eqldisplaybreak@prepen@-\@getpen{#1}\relax
1066   \fi
1067 }
```

TODO: describe

```
1068 \protected\def\eqldisplaybreak@measure{%
1069   \eq@ampprotect\eq@testopt@tight\eqldisplaybreak@measure@{4}%
1070 }
1071 \def\eqldisplaybreak@measure@#1{}
```

H.2 General Initialisation

TODO: describe

```
\eq@vspaceskip@ (skip) TODO: add a proper star variant?!
\eq@abovespace@ (skip)
\eq@belowspace@ (skip)
1072 \newskip\eq@vspaceskip@
1073 \newskip\eq@abovespace@
1074 \newskip\eq@belowspace@
1075 \let\eq@vspace@org\vspace
1076 \def\eq@vspace{\eq@ifstar@loose\eq@vspace@\eq@vspace@}
1077 \def\eq@vspace@#1{%
1078   \global\advance\eq@vspaceskip@\glueexpr#1\relax}
```

\eqldisplay@init

```
1079 \def\eqldisplay@init{%
1080   \eqldisplay@firstavail@\z@
1081   \eq@raisetag@firstlast@\z@
1082   \let\displaybreak\eqldisplaybreak@print
1083   \eqldisplaybreak@pen@\@MM
1084   \eq@vspaceskip@\z@skip
1085   \let\eq@vspace@org\vspace
1086   \let\vspace\eq@vspace
1087 }
```

\eqldisplay@close **TODO:** there seems to be an offset of 1em in predisplaysize towards actual content, nice.

TODO: must not use setlength or setcounter when calc is loaded

```
1088 \def\eqldisplay@close{%
1089   % \TODO temporary fix
1090   \ifdefined\eq@tagging@on
```

```

1091 \ifdefined\dollardollar@begin\else
1092 \belowdisplayskip-\belowdisplayskip
1093 \belowdisplayshortskip-\belowdisplayshortskip
1094 \fi
1095 \fi
1096 \ifdim\eqldisplay@firstavail@<\z@
1097 \eqldisplay@firstavail@\z@
1098 \fi
1099 \eql@skip@mode@leave@\z@
1100 \ifdim\eql@halign@prevdepth@=\maxdimen
1101 \ifdim\preplaysize=-\maxdimen
1102 \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1103 \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1104 \else
1105 \eql@skip@mode@above@\z@
1106 \eql@skip@mode@below@\z@
1107 \advance\eqldisplay@firstavail@\displayindent
1108 \ifdim\eqldisplay@firstavail@>\preplaysize
1109 \ifcase\eql@skip@mode@short\relax
1110 \or
1111 \eql@skip@mode@above@\@ne
1112 \or
1113 \eql@skip@mode@above@\@ne
1114 \ifnum\eql@totalrows@=\@ne
1115 \eql@skip@mode@below@\@ne
1116 \fi
1117 \or
1118 \eql@skip@mode@above@\@ne
1119 \eql@skip@mode@below@\@ne
1120 \fi
1121 \fi
1122 \fi
1123 \else
1124 \ifdim\eql@halign@prevdepth@=-\@m\p@
1125 \eql@skip@mode@above@\eql@skip@mode@top@above\relax
1126 \eql@skip@mode@below@\eql@skip@mode@top@below\relax
1127 \else
1128 \eql@skip@mode@above@\eql@skip@mode@par@above\relax
1129 \eql@skip@mode@below@\eql@skip@mode@par@below\relax
1130 \fi
1131 \fi
1132 \ifcase\eql@skip@mode@above@
1133 \or\or\or
1134 \predisplaypenalty\z@
1135 \or
1136 \predisplaypenalty\z@
1137 \fi
1138 \ifcase\eql@skip@mode@below@
1139 \or\or\or
1140 \eql@skip@mode@leave@\@ne
1141 \or
1142 \eql@skip@mode@leave@\tw@
1143 \fi
1144 \ifdefined\eql@skip@force@above
1145 \eql@skip@mode@above@\eql@skip@force@above\relax
1146 \fi
1147 \ifdefined\eql@skip@force@below
1148 \eql@skip@mode@below@\eql@skip@force@below\relax

```

```

1149 \fi
1150 \ifdefined\eq@skip@force@leave
1151   \eq@skip@mode@leave@eq@skip@force@leave\relax
1152 \fi
1153 \ifnum\eq@skip@mode@leave@>\z@
1154   \postdisplaypenalty\z@
1155 \fi
1156 \ifodd\eq@raisetag@firstlast@
1157   \ifcase\eq@skip@mode@above@
1158     \abovedisplayskip\glueexpr\eq@skip@tag@above\relax
1159   \or
1160     \abovedisplayskip\glueexpr\eq@skip@tag@above\relax
1161   \or
1162     \abovedisplayskip\glueexpr\eq@skip@tag@above\relax
1163   \or
1164     \abovedisplayskip\glueexpr\eq@skip@partag@above\relax
1165   \or
1166     \abovedisplayskip\glueexpr\eq@skip@partag@above\relax
1167   \or
1168     \abovedisplayskip\z@skip
1169   \or
1170     \abovedisplayskip\glueexpr\eq@skip@medtag@above\relax
1171   \or
1172     \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
1173 \fi
1174 \else
1175   \ifcase\eq@skip@mode@above@
1176     \abovedisplayskip\glueexpr\eq@skip@long@above\relax
1177   \or
1178     \abovedisplayskip\glueexpr\eq@skip@short@above\relax
1179   \or
1180     \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
1181   \or
1182     \abovedisplayskip\glueexpr\eq@skip@par@above\relax
1183   \or
1184     \abovedisplayskip\glueexpr\eq@skip@top@above\relax
1185   \or
1186     \abovedisplayskip\z@skip
1187   \or
1188     \abovedisplayskip\glueexpr\eq@skip@med@above\relax
1189   \or
1190     \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
1191 \fi
1192 \fi
1193 \ifnum\eq@raisetag@firstlast@>\@ne
1194   \ifcase\eq@skip@mode@below@
1195     \belowdisplayskip\glueexpr\eq@skip@tag@below\relax
1196   \or
1197     \belowdisplayskip\glueexpr\eq@skip@tag@below\relax
1198   \or
1199     \belowdisplayskip\glueexpr\eq@skip@tag@below\relax
1200   \or
1201     \belowdisplayskip\glueexpr\eq@skip@partag@below\relax
1202   \or
1203     \belowdisplayskip\glueexpr\eq@skip@partag@below\relax
1204   \or
1205     \belowdisplayskip\z@skip
1206 \or

```

```

1207     \belowdisplayskip\glueexpr\eq\skip@medtag@below\relax
1208     \or
1209     \belowdisplayskip\glueexpr\eq\skip@custom@below\relax
1210     \fi
1211 \else
1212     \ifcase\eq\skip@mode@below@
1213     \belowdisplayskip\glueexpr\eq\skip@long@below\relax
1214     \or
1215     \belowdisplayskip\glueexpr\eq\skip@short@below\relax
1216     \or
1217     \belowdisplayskip\glueexpr\eq\skip@cont@below\relax
1218     \or
1219     \belowdisplayskip\glueexpr\eq\skip@par@below\relax
1220     \or
1221     \belowdisplayskip\glueexpr\eq\skip@top@below\relax
1222     \or
1223     \belowdisplayskip\z\skip
1224     \or
1225     \belowdisplayskip\glueexpr\eq\skip@med@below\relax
1226     \or
1227     \belowdisplayskip\glueexpr\eq\skip@custom@below\relax
1228     \fi
1229 \fi
1230 \ifnum\eq\displaybreak@open@=\@MM\else
1231     \postdisplaypenalty\eq\displaybreak@open@
1232 \fi
1233 \ifnum\eq\displaybreak@prepen@=\@MM\else
1234     \predisplaypenalty\eq\displaybreak@prepen@
1235 \fi
1236 \advance\abovedisplayskip\eq\abovespace@
1237 \advance\belowdisplayskip\eq\belowspace@
1238 \advance\belowdisplayskip\eq\vspaceskip@
1239 \count@\prevgraf
1240 \advance\count@\eq\totalrows@
1241 \ifnum\count@>\z@
1242     \advance\count@\m@ne
1243 \fi
1244 \prevgraf\count@
1245 \global\eq\skip@mode@leave@\eq\skip@mode@leave@
1246 % \TODO temporary fix
1247 \ifdefined\eq\tagging@on
1248     \ifdefined\dollardollar@begin\else
1249         \belowdisplayskip-\belowdisplayskip
1250     \fi
1251 \fi
1252 }

1253 \def\eq\display@leave{%
1254     \let\label\eq\label@org
1255     \let\tag\eq\tag@default
1256     \let\raisetag\eq\raisetag@default
1257     \let\displaybreak\eq\displaybreak@default
1258     \let\intertext\eq\intertext@default
1259     \let\vspace\eq\vspace@org
1260 }

1261 \expandafter\def\expandafter\@arrayparboxrestore\expandafter{%
1262     \@arrayparboxrestore
1263     \eq\display@leave
1264     \ifdefined\eq\ampproof@active

```

```

1265 \eql@amprevert
1266 \fi
1267 \@displayfalse
1268 }

```

H.3 halign Support

TODO: describe

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

```

1269 \newbox\eql@strutbox@
1270 \def\eql@strut{\copy\eql@strutbox@}
1271 \let\eql@strut@field\eql@strut
1272 \let\eql@strut@tag\eql@strut
1273 \def\eql@strut@make{%
1274 \setbox\eql@strutbox@\hbox{%
1275 \@tempdima\normalbaselineskip
1276 \advance\@tempdima-\normallineskiplimit
1277 \@tempdimb.3\normalbaselineskip
1278 \advance\@tempdimb.5\normallineskiplimit
1279 \advance\@tempdima-\@tempdimb
1280 \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1281 }
1282 }
1283 \AtBeginDocument{\eql@strut@make}

```

TODO: describe **TODO:** note on “spread@equation

```

1284 \def\eql@halign@spread{%
1285 \eql@spread@\glueexpr\eql@spread@val\relax
1286 \advance\eql@spread@\normalbaselineskip
1287 \advance\eql@spread@-\baselineskip
1288 \ifdim\eql@spread@>\z@
1289 \openup\eql@spread@
1290 \ifdefined\spread@equation
1291 \let\spread@equation\@empty
1292 \fi
1293 \fi
1294 }

```

`gn@prevdepth@` (*dimen*)

```

1295 \newdimen\eql@halign@prevdepth@
1296 \def\eql@halign@catchprevdepth{%
1297 \ifvmode
1298 \eql@halign@prevdepth@\prevdepth
1299 \nointerlineskip
1300 \noindent
1301 \else
1302 \eql@halign@prevdepth@\maxdimen
1303 \fi
1304 }

1305 \def\eql@halign@leave{%
1306 \ifcase\eql@skip@mode@leave@
1307 \or

```



```

1308   \endgraf
1309   \or
1310   \endgraf
1311   \prevdepth-\@m\p@
1312   \fi
1313 }

```

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

```

1314 \def\eql@halign@before{%
1315   \ifdim\eql@halign@prevdepth@=\maxdimen\else
1316     \prevdepth\eql@halign@prevdepth@
1317   \fi
1318   \ifdim\prevdepth=-\@m\p@\else
1319     \ifdefined\eql@display@height
1320       \skip@\baselineskip
1321       \advance\skip@-\glueexpr\eql@display@height\relax
1322       \advance\skip@-\prevdepth\relax
1323       \ifdim\skip@<\lineskiplimit
1324         \skip@\lineskip
1325       \fi
1326       \advance\skip@-\eql@spread@\relax
1327       \vskip\skip@
1328       \nointerlineskip
1329     \else
1330       \vskip-\eql@spread@\relax
1331     \fi
1332   \fi
1333 }

```

TODO: describe

```

1334 \def\eql@halign@after{%
1335   \ifdefined\eql@display@depth
1336     \prevdepth\glueexpr\eql@display@depth\relax
1337   \fi
1338 }

```

TODO: describe

```

1339 \def\eql@halign@init#1{%
1340   \eql@halign@spread
1341   \eql@strut@make
1342   \everycr{\noalign{#1}}%
1343 }

```

H.4 Stack

TODO: describe

```

1344 \def\eql@stack@enable{%
1345   \let\eql@stack@save@single\eql@stack@save@single@
1346   \let\eql@stack@save@multi\eql@stack@save@multi@
1347   \let\eql@stack@save@boxed\eql@stack@save@boxed@
1348 }

```

TODO: describe

```

1349 \let\eql@stack@save@single\eql@stack@enable
1350 \let\eql@stack@save@multi\eql@stack@enable

```

```

1351 \let\eqL@stack@save@boxed\eqL@stack@enable
1352 \let\eqL@stack@restore\@empty

```

TODO: describe

```

1353 \def\eqL@stack@save@reg#1{\global#1\the#1\relax}
1354 \def\eqL@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

TODO: describe

```

1355 \def\eqL@stack@save@single@{%
1356   \let\eqL@stack@nextlabel\eqL@nextlabel
1357   \let\eqL@stack@nexttag\eqL@nexttag
1358   \edef\eqL@stack@restore{%
1359     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1360     \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1361     \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1362     \eqL@stack@save@reg\eqL@displaybreak@pen@
1363     \eqL@stack@save@reg\eqL@vspaceskip@
1364     \eqL@stack@save@reg\eqL@shape@pos@
1365     \eqL@stack@save@reg\eqL@shape@amount@
1366     \eqL@stack@save@reg\eqL@display@firstavail@
1367     \eqL@stack@save@reg\eqL@raisetag@amount@
1368     \eqL@stack@save@reg\eqL@raisetag@firstlast@
1369   }%
1370 }

```

TODO: describe

```

1371 \def\eqL@stack@save@multi@{%
1372   \let\eqL@stack@nextlabel\eqL@nextlabel
1373   \let\eqL@stack@nexttag\eqL@nexttag
1374   \let\eqL@stack@tagwidth@tab\eqL@tagwidth@tab
1375   \let\eqL@stack@fieldlength@tab\eqL@fieldlength@tab
1376   \let\eqL@stack@colwidth@tab\eqL@colwidth@tab
1377   \let\eqL@stack@label@thepage\eqL@label@thepage
1378   \let\eqL@stack@label@currentHref\eqL@label@currentHref
1379   \edef\eqL@stack@restore{%
1380     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1381     \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1382     \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1383     \eqL@stack@save@let\eqL@stack@tagwidth@tab\eqL@tagwidth@tab
1384     \eqL@stack@save@let\eqL@stack@fieldlength@tab\eqL@fieldlength@tab
1385     \eqL@stack@save@let\eqL@stack@colwidth@tab\eqL@colwidth@tab
1386     \eqL@stack@save@let\eqL@stack@label@thepage\eqL@label@thepage
1387     \eqL@stack@save@let\eqL@stack@label@currentHref\eqL@label@currentHref
1388     \eqL@stack@save@reg\eqL@displaybreak@pen@
1389     \eqL@stack@save@reg\eqL@vspaceskip@
1390     \eqL@stack@save@reg\eqL@shape@pos@
1391     \eqL@stack@save@reg\eqL@shape@amount@
1392     \eqL@stack@save@reg\eqL@display@firstavail@
1393     \eqL@stack@save@reg\eqL@raisetag@amount@
1394     \eqL@stack@save@reg\eqL@raisetag@firstlast@
1395     \eqL@stack@save@reg\eqL@column@
1396     \eqL@stack@save@reg\eqL@totalcolumns@
1397     \eqL@stack@save@reg\eqL@line@avail@
1398     \eqL@stack@save@reg\eqL@line@pos@
1399     \eqL@stack@save@reg\eqL@line@width@
1400     \eqL@stack@save@reg\eqL@line@depth@
1401     \eqL@stack@save@reg\eqL@line@height@
1402     \eqL@stack@save@reg\eqL@tagwidth@max@

```

```

1403 \eql@stack@save@reg\eql@numbering@target@
1404 \eql@stack@save@reg\eql@row@
1405 \eql@stack@save@reg\eql@tagrows@
1406 }%
1407 }
1408 \def\eql@stack@save@boxed@{%
1409 \edef\eql@stack@restore{%
1410 \eql@stack@save@reg\eql@row@
1411 \eql@stack@save@reg\eql@totalrows@
1412 \eql@stack@save@reg\eql@shape@pos@
1413 \eql@stack@save@reg\eql@shape@amount@
1414 }%
1415 }

```

I Horizontal Spacing for Lines

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

I.1 Supporting Definitionss

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

```

1416 \ifdefined\inf@bad\else%
1417 \newcount\inf@bad
1418 \inf@bad1000000\relax
1419 \fi

```

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

```

1420 \let\eql@restore@hfuzz\@empty
1421 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}

```

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

```

1422 \newcount\eql@shape@pos@
1423 \newdimen\eql@shape@amount@

```

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

```

1424 \newdimen\eql@marginleft@
1425 \newdimen\eql@marginright@

```

`\eql@marginbadness@` The registers `\eql@marginbadness@` and `\eql@maxbadness@` 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:

```

1426 \newcount\eql@marginbadness@
1427 \newcount\eql@maxbadness@
1428 \eql@marginbadness@\inf@bad
1429 \eql@maxbadness@\inf@bad

```

I.2 Shape Schemes

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

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

```
1430 \def\eql@shape@tab@default{default}
1431 \def\eql@shape@tab@left{left}
1432 \def\eql@shape@tab@center{center}
1433 \def\eql@shape@tab@right{right}
1434 \def\eql@shape@tab@first{first}
1435 \def\eql@shape@tab@hanging{hanging}
1436 \def\eql@shape@tab@steps{steps}
```

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

```
1437 \let\eql@shape@tab@def\eql@shape@tab@default
1438 \let\eql@shape@tab@\eql@shape@tab@default
1439 \let\eql@shape@tab@l\eql@shape@tab@left
1440 \let\eql@shape@tab@c\eql@shape@tab@center
1441 \let\eql@shape@tab@r\eql@shape@tab@right
1442 \let\eql@shape@tab@rc\eql@shape@tab@first
1443 \let\eql@shape@tab@indent\eql@shape@tab@first
1444 \let\eql@shape@tab@hang\eql@shape@tab@hanging
1445 \let\eql@shape@tab@lc\eql@shape@tab@hanging
1446 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
1447 \let\eql@shape@tab@lcr\eql@shape@tab@steps
```

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

```
1448 \let\eql@shape@mode\eql@shape@tab@default
```

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

```
1449 \def\eql@shape@set#1{%
1450   \ifcsname eql@shape@tab@#1\endcsname
1451     \expandafter\let\expandafter\eql@shape@mode
1452     \csname eql@shape@tab@#1\endcsname
1453   \else
1454     \eql@error{shape ‘#1’ unknown: setting to default}%
1455     \let\eql@shape@mode\eql@shape@tab@default
1456   \fi
1457 }
```

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

```
1458 \def\eql@shape@layoutcenter@left{\eql@shape@pos@z@\eql@shape@amount@z@}
1459 \def\eql@shape@layoutcenter@center{\eql@shape@pos@\@ne\eql@shape@amount@z@}
1460 \def\eql@shape@layoutcenter@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
1461 \let\eql@shape@layoutcenter@default\eql@shape@layoutcenter@center
1462 \def\eql@shape@layoutleft@left{\eql@shape@pos@z@\eql@shape@amount@z@}
1463 \def\eql@shape@layoutleft@center{\eql@shape@pos@\@ne\eql@shape@amount@z@}
1464 \def\eql@shape@layoutleft@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
1465 \let\eql@shape@layoutleft@default\eql@shape@layoutleft@left
```

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

```

1466 \def\eqL@shape@layoutcenter@first{%
1467   \eqL@shape@pos@z@
1468   \eqL@shape@amount@z@
1469   \ifnum\eqL@totalrows@>\@ne
1470     \ifnum\eqL@row@=\@ne
1471       \eqL@shape@amount@\eqL@indent@
1472     \fi
1473   \fi
1474 }
1475 \def\eqL@shape@layoutleft@first{%
1476   \eqL@shape@pos@z@
1477   \eqL@shape@amount@z@
1478   \ifnum\eqL@totalrows@>\@ne
1479     \ifnum\eqL@row@=\@ne
1480       \eqL@shape@amount@\eqL@indent@
1481     \fi
1482   \fi
1483 }

```

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:

```

1484 \def\eqL@shape@layoutcenter@hanging{%
1485   \eqL@shape@pos@z@
1486   \eqL@shape@amount@\eqL@indent@
1487   \ifnum\eqL@totalrows@>\@ne
1488     \ifnum\eqL@row@=\@ne
1489       \eqL@shape@amount@z@
1490     \fi
1491   \fi
1492 }
1493 \def\eqL@shape@layoutleft@hanging{%
1494   \eqL@shape@pos@z@
1495   \eqL@shape@amount@z@
1496   \ifnum\eqL@totalrows@>\@ne
1497     \ifnum\eqL@row@=\@ne
1498       \eqL@shape@amount@-\eqL@indent@
1499     \fi
1500   \fi
1501 }

```

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:

```

1502 \def\eqL@shape@layoutcenter@steps{%
1503   \eqL@shape@amount@z@
1504   \eqL@shape@pos@one
1505   \ifnum\eqL@totalrows@>\@ne
1506     \ifnum\eqL@row@=\@ne
1507       \eqL@shape@pos@z@
1508     \fi
1509     \ifnum\eqL@row@=\eqL@totalrows@
1510       \eqL@shape@pos@tw@
1511     \fi
1512   \fi
1513 }
1514 \def\eqL@shape@layoutleft@steps{%

```

```

1515 \eql@shape@pos@\z@
1516 \eql@shape@amount@\z@
1517 \ifnum\eql@totalrows@>\@ne
1518   \ifnum\eql@row@=\@ne
1519     \eql@shape@amount@-\eql@indent@
1520     \fi
1521   \ifnum\eql@row@=\eql@totalrows@
1522     \eql@shape@amount@\eql@indent@
1523     \fi
1524   \fi
1525 }

```

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

```

1526 \let\eql@shape@eval\undefined
1527 \def\eql@shape@sel{%
1528   \expandafter\let\expandafter\eql@shape@eval
1529     \csname eql@shape%
1530     @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
1531     @\eql@shape@mode\endcsname
1532 }

```

`\eql@adjust@alignleft` Adjust the alignment of the current equation line. For left alignment an optional argument `\eql@adjust@alignright` specifies the amount of indentation:

`\eql@adjust@aligncenter`

```

1533 \def\eql@adjust@alignleft{%
1534   \global\eql@shape@pos@\z@
1535   \eql@srbgroup\eql@ifstar@tight
1536   {\eql@adjust@alignleft@[\eql@indent@]}%
1537   {\eql@ifnextgobble@tight{!}}%
1538   {\eql@adjust@alignleft@[-\eql@indent@]}%
1539   {\eql@testopt@tight\eql@adjust@alignleft@\z@}%
1540   }%
1541 }
1542 \def\eql@adjust@alignleft@[#1]{%
1543   \eql@sregroup\global\eql@shape@amount@\glueexpr#1\relax}
1544 \def\eql@adjust@aligncenter{%
1545   \global\eql@shape@pos@\@ne\global\eql@shape@amount@\z@}
1546 \def\eql@adjust@alignright{%
1547   \global\eql@shape@pos@\tw@\global\eql@shape@amount@\z@}

```

I.3 Adjustment Methods

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

```

1548 \def\eql@adjust@try#1{%
1549   \setbox\eql@fieldbox@\hbox to\displaywidth{%
1550     \unhbox\eql@fieldbox@\unkern\kern#1}%
1551 }

```

`\eql@adjust@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`:

```

1552 \def\eql@adjust@print#1#2{%
1553   \eql@restore@hfuzz

```

```

1554 \hbox to\displaywidth{%
1555   #1%
1556   \unhbox\eql@fieldbox@\unkern
1557   #2%
1558   \eql@tagging@mathaddlast
1559 }%
1560 }

```

`\just@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:

```

1561 \def\eql@adjust@print@alignleft#1#2{%
1562   \ifnum\eql@row@=\@one
1563     \global\eql@display@firstavail@#1%
1564   \fi
1565   \eql@adjust@print{\kern#1}{\kern#2}%
1566 }
1567 \def\eql@adjust@print@alignright#1#2{%
1568   \ifnum\eql@row@=\@one
1569     \eql@display@firstavail@\displaywidth
1570     \advance\eql@display@firstavail@-\eql@fieldwidth@
1571     \global\advance\eql@display@firstavail@-#2%
1572   \fi
1573   \eql@adjust@print{\kern#1\hfil}{\unskip\kern#2}%
1574 }
1575 \def\eql@adjust@print@aligncenter#1#2{%
1576   \ifnum\eql@row@=\@one
1577     \eql@display@firstavail@\displaywidth
1578     \advance\eql@display@firstavail@-\eql@fieldwidth@
1579     \advance\eql@display@firstavail@#1%
1580     \advance\eql@display@firstavail@-#2%
1581     \global\divide\eql@display@firstavail@\tw@
1582   \fi
1583   \eql@adjust@print{\kern#1\hfil}{\kern#2}%
1584 }

```

`\eql@adjust@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:

```

1585 \def\eql@adjust@init{%
1586   \eql@save@hfuzz
1587   \hfuzz\maxdimen
1588   \eql@shape@sel
1589 }

```

`\eql@adjust@sel@tag` Select the appropriate adjustment method depending on the selected layout, selected tag placement, current alignment position and on whether a tag is present or not:

```

1590 \def\eql@adjust@sel@tag{%
1591   \eql@tagging@tagaddbox
1592   \ifcase\eql@shape@pos@
1593     \eql@tagging@alignleft
1594   \or
1595     \eql@tagging@aligncenter
1596   \or
1597     \eql@tagging@alignright

```

```

1598 \fi
1599 \csname eql@adjust%
1600   @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
1601   @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1602   @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
1603   @tag\endcsname
1604 }
1605 \def\eql@adjust@sel@notag{%
1606   \eql@tagging@tagaddbox
1607   \ifcase\eql@shape@pos@
1608     \eql@tagging@alignleft
1609   \or
1610     \eql@tagging@aligncenter
1611   \or
1612     \eql@tagging@alignright
1613   \fi
1614   \csname eql@adjust%
1615     @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
1616     @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1617     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
1618     @notag\endcsname
1619 }

```

`\eql@adjust@calc` **TODO:** any init needed for left alignment layout? `marginleft` is used per line!

```

1620 \def\eql@adjust@calc{%
1621   \ifdefined\eql@layoutleft
1622     \eql@layoutleftmargin@\glueexpr\eql@layoutleftmargin@val\relax
1623   \else
1624     \eql@columns@inter@\z@
1625     \eql@adjust@tagmargin
1626     \ifdefined\eql@paddingmax
1627       \eql@marginleft@\z@
1628       \eql@marginright@\z@
1629     \else
1630       \dimen@\displaywidth
1631       \advance\dimen@-\eql@totalwidth@
1632       \advance\dimen@-\eql@tagmargin@
1633       \divide\dimen@\tw@
1634       \eql@marginleft@\dimen@
1635       \advance\eql@marginleft@-\glueexpr\eql@paddingleft@val\relax
1636       \ifdim\eql@marginleft@<\z@
1637         \eql@marginleft@\z@
1638       \fi
1639       \eql@marginright@\dimen@
1640       \advance\eql@marginright@-\glueexpr\eql@paddingright@val\relax
1641       \ifdim\eql@marginright@<\z@
1642         \eql@marginright@\z@
1643       \fi
1644     \fi
1645   \fi
1646 }

```

I.4 Central Alignment Layout

TODO: describe

TODO: check all these!!


```

1647 \def\eql@adjust@layoutcenter@tagsright@aligncenter@notag{%
1648   \dimen@ \displaywidth
1649   \advance\dimen@-\eql@fieldwidth@
1650   \ifdim\dimen@>\eql@tagmargin@
1651     \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1652   \else
1653     \eql@adjust@print@alignleft\z@\z@
1654   \fi
1655 }

```

TODO: describe

```

1656 \def\eql@adjust@layoutcenter@tagsright@aligncenter@tag{%
1657   \dimen@ \displaywidth
1658   \ifdim\eql@tagwidth@<\eql@tagmargin@
1659     \advance\dimen@-\eql@tagmargin@
1660   \else
1661     \advance\dimen@-2\eql@tagwidth@
1662     \advance\dimen@\eql@tagmargin@
1663   \fi
1664   \ifdim\eql@fieldwidth@<\dimen@
1665     \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1666     \eql@tagbox@print@tagsright
1667   \else
1668     \eql@adjust@try\eql@tagwidth@
1669     \ifnum\badness<\eql@maxbadness@
1670       \ifdim\eql@tagwidth@<\eql@tagmargin@
1671         \eql@adjust@print@alignleft\z@\eql@tagwidth@
1672       \else
1673         \eql@adjust@print@alignright\z@\eql@tagwidth@
1674       \fi
1675     \eql@tagbox@print@tagsright
1676   \else
1677     \eql@adjust@layoutcenter@tagsright@aligncenter@notag
1678     \eql@tagbox@print@tagsright@raise
1679   \fi
1680 \fi
1681 }

```

TODO: describe

```

1682 \def\eql@adjust@layoutcenter@tagsleft@aligncenter@notag{%
1683   \dimen@ \displaywidth
1684   \advance\dimen@-\eql@tagmargin@
1685   \ifdim\eql@fieldwidth@<\dimen@
1686     \eql@adjust@print@aligncenter\eql@tagmargin@\z@
1687   \else
1688     \eql@adjust@print@alignright\z@\z@
1689   \fi
1690 }

```

TODO: describe

```

1691 \def\eql@adjust@layoutcenter@tagsleft@aligncenter@tag{%
1692   \dimen@ \displaywidth
1693   \ifdim\eql@tagwidth@<\eql@tagmargin@
1694     \advance\dimen@-\eql@tagmargin@
1695   \else
1696     \advance\dimen@-2\eql@tagwidth@
1697     \advance\dimen@\eql@tagmargin@
1698   \fi

```

```

1699 \ifdim\eq@fieldwidth@<\dimen@
1700   \eq@tagbox@print@tagsleft
1701   \eq@adjust@print@aligncenter\eq@tagmargin@z@
1702 \else
1703   \eq@adjust@try\eq@tagwidth@
1704   \ifnum\badness<\eq@maxbadness@
1705     \eq@tagbox@print@tagsleft
1706     \ifdim\eq@tagwidth@<\eq@tagmargin@
1707       \eq@adjust@print@alignright\eq@tagwidth@z@
1708     \else
1709       \eq@adjust@print@alignleft\eq@tagwidth@z@
1710     \fi
1711   \else
1712     \eq@tagbox@print@tagsleft@raise
1713     \eq@adjust@layoutcenter@tagsleft@aligncenter@notag
1714   \fi
1715 \fi
1716 \eq@display@firstavail@setz@
1717 }

```

TODO: describe

```

1718 \def\eq@adjust@layoutcenter@tagsright@alignleft@notag{%
1719   \dimen@\displaywidth
1720   \advance\dimen@-\eq@marginleft@
1721   \advance\dimen@-\eq@shape@amount@
1722   \ifdim\eq@fieldwidth@<\dimen@
1723     \dimen@\eq@marginleft@
1724     \advance\dimen@\eq@shape@amount@
1725     \eq@adjust@print@alignleft\dimen@z@
1726   \else
1727     \eq@adjust@print@alignrightz@z@
1728   \fi
1729 }

```

TODO: describe

```

1730 \def\eq@adjust@layoutcenter@tagsright@alignleft@tag{%
1731   \dimen@\eq@marginleft@
1732   \advance\dimen@\eq@shape@amount@
1733   \advance\dimen@\eq@tagwidth@
1734   \eq@adjust@try\dimen@
1735   \ifnum\badness<\eq@marginbadness@
1736     \dimen@\eq@marginleft@
1737     \advance\dimen@\eq@shape@amount@
1738     \eq@adjust@print@alignleft\dimen@\eq@tagwidth@
1739     \eq@tagbox@print@tagsright
1740   \else
1741     \ifdim\eq@marginleft@>-\eq@shape@amount@
1742       \eq@adjust@try\eq@tagwidth@
1743     \fi
1744     \ifnum\badness<\eq@maxbadness@
1745       \eq@adjust@print@alignrightz@\eq@tagwidth@
1746       \eq@tagbox@print@tagsright
1747     \else
1748       \eq@adjust@layoutcenter@tagsright@alignleft@notag
1749       \eq@tagbox@print@tagsright@raise
1750     \fi
1751   \fi
1752 }

```

TODO: describe

```
1753 \def\eq@adjust@layoutcenter@tagsleft@alignright@notag{%
1754   \dimen@ \displaywidth
1755   \advance\dimen@-\eq@tagmargin@
1756   \advance\dimen@-\eq@marginright@
1757   \ifdim\eq@fieldwidth@<\dimen@
1758     \eq@adjust@print@alignright\z@\eq@marginright@
1759   \else
1760     \eq@adjust@print@alignleft\z@\z@
1761   \fi
1762 }
```

TODO: describe

```
1763 \def\eq@adjust@layoutcenter@tagsleft@alignright@tag{%
1764   \dimen@ \eq@marginright@
1765   \advance\dimen@ \eq@tagwidth@
1766   \eq@adjust@try\dimen@
1767   \ifnum\badness<\eq@marginbadness@
1768     \eq@tagbox@print@tagsleft
1769     \eq@adjust@print@alignright\eq@tagwidth@\eq@marginright@
1770   \else
1771     \ifdim\eq@marginright@>\z@
1772       \eq@adjust@try\eq@tagwidth@
1773     \fi
1774     \ifnum\badness<\eq@maxbadness@
1775       \eq@tagbox@print@tagsleft
1776       \eq@adjust@print@alignleft\eq@tagwidth@\z@
1777     \else
1778       \eq@tagbox@print@tagsleft@raise
1779       \eq@adjust@layoutcenter@tagsleft@alignright@notag
1780     \fi
1781   \fi
1782   \eq@display@firstavail@set\z@
1783 }
```

TODO: describe

```
1784 \def\eq@adjust@layoutcenter@tagsright@alignright@notag{%
1785   \dimen@ \displaywidth
1786   \advance\dimen@-\eq@tagmargin@
1787   \advance\dimen@-\eq@marginright@
1788   \ifdim\eq@fieldwidth@<\dimen@
1789     \dimen@ \eq@tagmargin@
1790     \advance\dimen@ \eq@marginright@
1791     \eq@adjust@print@alignright\z@\dimen@
1792   \else
1793     \eq@adjust@print@alignleft\z@\z@
1794   \fi
1795 }
```

TODO: describe

```
1796 \def\eq@adjust@layoutcenter@tagsright@alignright@tag{%
1797   \dimen@ \eq@tagmargin@
1798   \advance\dimen@ \eq@marginright@
1799   \ifdim\eq@tagwidth@<\dimen@
1800     \eq@adjust@try\dimen@%
1801     \ifnum\badness<\eq@marginbadness@
1802       \eq@adjust@print@alignright\z@\dimen@
```

```

1803     \eql@tagbox@print@tagsright
1804   \else
1805     \eql@adjust@try\eql@tagwidth@
1806     \ifnum\badness<\eql@maxbadness@
1807       \eql@adjust@print@alignleft\z@\eql@tagwidth@
1808       \eql@tagbox@print@tagsright
1809     \else
1810       \eql@adjust@print@alignleft\z@\z@
1811       \eql@tagbox@print@tagsleft@raise
1812     \fi
1813   \fi
1814 \else
1815   \eql@adjust@try\eql@tagwidth@
1816   \ifnum\badness<\eql@maxbadness@
1817     \eql@adjust@print@alignright\z@\eql@tagwidth@
1818     \eql@tagbox@print@tagsright
1819   \else
1820     \eql@adjust@layoutcenter@tagsright@alignright@notag
1821     \eql@tagbox@print@tagsright@raise
1822   \fi
1823 \fi
1824 }

```

TODO: describe

```

1825 \def\eql@adjust@layoutcenter@tagsleft@alignleft@notag{%
1826   \dimen@\displaywidth
1827   \advance\dimen@-\eql@tagmargin@
1828   \advance\dimen@-\eql@marginleft@
1829   \advance\dimen@-\eql@shape@amount@
1830   \ifdim\eql@fieldwidth@<\dimen@
1831     \dimen@\eql@tagmargin@
1832     \advance\dimen@\eql@marginleft@
1833     \advance\dimen@\eql@shape@amount@
1834     \eql@adjust@print@alignleft\dimen@\z@
1835   \else
1836     \eql@adjust@print@alignright\z@\z@
1837   \fi
1838 }

```

TODO: describe

```

1839 \def\eql@adjust@layoutcenter@tagsleft@alignleft@tag{%
1840   \dimen@\eql@tagmargin@
1841   \advance\dimen@\eql@marginleft@
1842   \advance\dimen@\eql@shape@amount@
1843   \ifdim\eql@tagwidth@<\dimen@
1844     \eql@adjust@try\dimen@%
1845     \ifnum\badness<\eql@marginbadness@
1846       \eql@tagbox@print@tagsleft
1847       \eql@adjust@print@alignleft\dimen@\z@
1848     \else
1849       \eql@adjust@try\eql@tagwidth@
1850       \ifnum\badness<\eql@maxbadness@
1851         \eql@tagbox@print@tagsleft
1852         \eql@adjust@print@alignright\eql@tagwidth@\z@
1853       \else
1854         \eql@tagbox@print@tagsleft@raise
1855         \eql@adjust@print@alignright\z@\z@
1856       \fi

```

```

1857 \fi
1858 \else
1859 \eql@adjust@try\eql@tagwidth@
1860 \ifnum\badness<\eql@maxbadness@
1861 \eql@tagbox@print@tagsleft
1862 \eql@adjust@print@alignleft\eql@tagwidth@\z@
1863 \else
1864 \eql@tagbox@print@tagsleft@raise
1865 \eql@adjust@layoutcenter@tagsleft@alignleft@notag
1866 \fi
1867 \fi
1868 \eql@display@firstavail@set\z@
1869 }

```

eql@adjust@tagmargin

```

1870 \def\eql@adjust@tagmargin{%
1871 \ifdefined\eql@tagmargin@val
1872 \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
1873 \else
1874 \eql@tagmargin@\eql@tagwidth@max@
1875 \ifdim\eql@tagmargin@>\z@
1876 \advance\eql@tagmargin@-\eql@tagsepmin@
1877 \fi
1878 \fi

1879 \dimen@1\eql@tagrows@1p@
1880 \ifnum\eql@totalrows@=\@ne
1881 \ifnum\eql@tagrows@=\@ne
1882 \advance\dimen@1sp\relax
1883 \fi
1884 \fi

1885 \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
1886 \eql@tagmargin@\z@
1887 \fi

1888 \@tempdima\displaywidth
1889 \advance\@tempdima-\eql@totalwidth@
1890 \advance\@tempdima-\eql@columns@inter@\eql@colsepmin@
1891 \@tempdimb\@tempdima
1892 \advance\@tempdimb-\tw@\eql@tagmargin@
1893 \ifdim\@tempdimb>\z@
1894 \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
1895 \eql@tagmargin@\z@
1896 \fi
1897 \fi
1898 }

```

I.5 Left Alignment Layout

TODO: describe

```

1899 \def\eql@adjust@layoutleft@alignleft{%
1900 \eql@marginleft@\eql@layoutleftmargin@
1901 \advance\eql@marginleft@\eql@shape@amount@
1902 \ifdim\eql@marginleft@<\eql@layoutleftmarginmin@
1903 \eql@marginleft@\eql@layoutleftmarginmin@
1904 \fi

```

```

1905 \ifdim\eq@marginleft@>\eq@layoutleftmarginmax@
1906   \eq@marginleft@\eq@layoutleftmarginmax@
1907   \fi
1908 }

```

TODO: perform checks based on unstretched dimension?! **TODO:** mention alternatives to fill; emphasis is on good left margin with ragged right (allow space between tag and equation in close case)

```

1909 \def\eq@adjust@layoutleft@alignleft@notag{%
1910   \ifdim\eq@layoutleftmarginmin@<\eq@marginleft@
1911     \eq@adjust@try\eq@marginleft@
1912     \ifnum\badness<\eq@marginbadness@
1913       \eq@adjust@print@alignleft\eq@marginleft@%z@
1914     \else
1915       \eq@adjust@print@alignleft\eq@layoutleftmarginmin@%z@
1916     \fi
1917   \else
1918     \eq@adjust@print@alignleft\eq@marginleft@%z@
1919   \fi
1920 }

```

TODO: describe

```

1921 \def\eq@adjust@layoutleft@tagsright@alignleft@notag{%
1922   \eq@adjust@layoutleft@alignleft
1923   \eq@adjust@layoutleft@alignleft@notag
1924 }
1925 \let\eq@adjust@layoutleft@tagsleft@alignleft@notag
1926   \eq@adjust@layoutleft@tagsright@alignleft@notag

```

TODO: what is worse, extend into margin or raise tag? this assumes raise tag, but other option might be better **TODO:** mention alternatives to fill; emphasis is on good left margin with ragged right (allow space between tag and equation in close case)

```

1927 \def\eq@adjust@layoutleft@tagsright@alignleft@tag{%
1928   \eq@adjust@layoutleft@alignleft
1929   \dimen@\eq@marginleft@
1930   \advance\dimen@\eq@tagwidth@
1931   \eq@adjust@try\dimen@
1932   \ifnum\badness<\eq@marginbadness@
1933     \eq@adjust@print@alignleft\eq@marginleft@\eq@tagwidth@
1934     \eq@tagbox@print@tagsright
1935   \else
1936     \ifdim\eq@layoutleftmarginmin@<\eq@marginleft@
1937       \dimen@\eq@layoutleftmarginmin@
1938       \advance\dimen@\eq@tagwidth@
1939       \eq@adjust@try\dimen@
1940     \fi
1941     \ifnum\badness<\eq@maxbadness@
1942       \eq@adjust@print@alignleft\eq@layoutleftmarginmin@\eq@tagwidth@
1943       \eq@tagbox@print@tagsright
1944     \else
1945       \eq@adjust@layoutleft@alignleft@notag
1946       \eq@tagbox@print@tagsright@raise
1947     \fi
1948   \fi
1949 }

```

```

1950 \def\eq@adjust@layoutleft@tagsleft@alignleft@tag{%

```

```

1951 \eql@adjust@layoutleft@alignleft
1952 \ifdim\eql@tagwidth@<\eql@layoutleftmarginmin@
1953   \eql@tagbox@print@tagsleft
1954   \eql@adjust@layoutleft@notag
1955 \else
1956   \ifdim\eql@tagwidth@<\eql@marginleft@
1957     \eql@adjust@try\eql@marginleft@
1958     \ifnum\badness<\eql@marginbadness@
1959       \eql@tagbox@print@tagsleft
1960       \eql@adjust@print@alignleft\eql@marginleft@ \z@
1961     \else
1962       \eql@adjust@try\eql@tagwidth@
1963       \ifnum\badness<\eql@maxbadness@
1964         \eql@tagbox@print@tagsleft
1965         \eql@adjust@print@alignleft\eql@tagwidth@ \z@
1966       \else
1967         \eql@tagbox@print@tagsleft@raise
1968         \eql@adjust@print@alignleft\eql@layoutleftmarginmin@ \z@
1969       \fi
1970     \fi
1971   \else
1972     \ifdim\eql@tagwidth@>\eql@layoutleftmarginmax@
1973       \eql@tagbox@print@tagsleft@raise
1974       \eql@adjust@layoutleft@alignleft@notag
1975     \else
1976       \eql@adjust@try\eql@tagwidth@
1977       \ifnum\badness<\eql@maxbadness@
1978         \eql@tagbox@print@tagsleft
1979         \eql@adjust@print@alignleft\eql@tagwidth@ \z@
1980       \else
1981         \eql@tagbox@print@tagsleft@raise
1982         \eql@adjust@layoutleft@alignleft@notag
1983       \fi
1984     \fi
1985   \fi
1986 \fi
1987 \eql@display@firstavail@set \z@
1988 }

```

TODO: describe

```

1989 \def\eql@adjust@layoutleft@alignright@notag{%
1990   \eql@marginleft@\eql@layoutleftmargin@
1991   \ifdim\eql@layoutleftmarginmin@<\eql@marginleft@
1992     \eql@adjust@try\eql@marginleft@
1993     \ifnum\badness<\eql@marginbadness@
1994       \eql@adjust@print@alignright\eql@marginleft@ \z@
1995     \else
1996       \eql@adjust@print@alignright\eql@layoutleftmarginmin@ \z@
1997     \fi
1998   \else
1999     \eql@adjust@print@alignright\eql@marginleft@ \z@
2000   \fi
2001 }
2002 \let\eql@adjust@layoutleft@tagsright@alignright@notag
2003 \eql@adjust@layoutleft@alignright@notag
2004 \let\eql@adjust@layoutleft@tagsleft@alignright@notag
2005 \eql@adjust@layoutleft@alignright@notag

```

TODO: describe

```
2006 \def\eq@adjust@layoutleft@tagsright@alignright@tag{%
2007   \dimen@eq@marginleft@
2008   \advance\dimen@eq@tagwidth@
2009   \eq@adjust@try\dimen@
2010   \ifnum\badness<\eq@marginbadness@
2011     \eq@adjust@print@alignright\eq@marginleft\eq@tagwidth@
2012     \eq@tagbox@print@tagsright
2013   \else
2014     \ifdim\eq@layoutleftmarginmin@<\eq@marginleft@
2015       \dimen@eq@layoutleftmarginmin@
2016       \advance\dimen@eq@tagwidth@
2017       \eq@adjust@try\dimen@
2018     \fi
2019     \ifnum\badness<\eq@maxbadness@
2020       \eq@adjust@print@alignright\eq@layoutleftmarginmin\eq@tagwidth@
2021       \eq@tagbox@print@tagsright
2022     \else
2023       \eq@adjust@layoutleft@alignright@notag
2024       \eq@tagbox@print@tagsright@raise
2025     \fi
2026   \fi
2027 }
```

TODO: describe

```
2028 \def\eq@adjust@layoutleft@tagsleft@alignright@tag{%
2029   \ifdim\eq@tagwidth@<\eq@layoutleftmarginmin@
2030     \eq@tagbox@print@tagsleft
2031     \eq@adjust@layoutleft@alignright@notag
2032   \else
2033     \ifdim\eq@tagwidth@<\eq@marginleft@
2034       \eq@adjust@try\eq@marginleft@
2035       \ifnum\badness<\eq@marginbadness@
2036         \eq@tagbox@print@tagsleft
2037         \eq@adjust@print@alignright\eq@marginleft\z@
2038       \else
2039         \eq@adjust@try\eq@tagwidth@
2040         \ifnum\badness<\eq@maxbadness@
2041           \eq@tagbox@print@tagsleft
2042           \eq@adjust@print@alignright\eq@tagwidth\z@
2043         \else
2044           \eq@tagbox@print@tagsleft@raise
2045           \eq@adjust@print@alignright\eq@layoutleftmarginmin\z@
2046         \fi
2047       \fi
2048     \else
2049       \ifdim\eq@tagwidth@>\eq@layoutleftmarginmax@
2050         \eq@tagbox@print@tagsleft@raise
2051         \eq@adjust@layoutleft@alignright@notag
2052       \else
2053         \eq@adjust@try\eq@tagwidth@
2054         \ifnum\badness<\eq@maxbadness@
2055           \eq@tagbox@print@tagsleft
2056           \eq@adjust@print@alignright\eq@tagwidth\z@
2057         \else
2058           \eq@tagbox@print@tagsleft@raise
2059           \eq@adjust@layoutleft@alignright@notag
```



```

2060     \fi
2061     \fi
2062     \fi
2063     \fi
2064     \eql@display@firstavail@set\z@
2065 }

2066 \def\eql@adjust@layoutleft@aligncenter{%
2067   \eql@error{shove center not implemented for left alignment}%
2068 }
2069 \let\eql@adjust@layoutleft@tagsright@aligncenter@notag
2070   \eql@adjust@layoutleft@aligncenter
2071 \let\eql@adjust@layoutleft@tagsright@aligncenter@tag
2072   \eql@adjust@layoutleft@aligncenter
2073 \let\eql@adjust@layoutleft@tagsleft@aligncenter@notag
2074   \eql@adjust@layoutleft@aligncenter
2075 \let\eql@adjust@layoutleft@tagsleft@aligncenter@tag
2076   \eql@adjust@layoutleft@aligncenter

```

J Multi-Line Support

TODO: describe

J.1 Table Registers

TODO: can we unite `\eql@fieldlength@tab` and `\eql@tagwidth@tab` **TODO:** then process sequentially not using `ifcase`?

```

\eql@fieldlength@tab
\eql@fieldlength@save
\eql@fieldlength@get
2077 \let\eql@fieldlength@tab\@empty
2078 \def\eql@fieldlength@save#1{%
2079   \begingroup
2080     \let\or\relax
2081     \global\edef\eql@fieldlength@tab{%
2082       \eql@fieldlength@tab
2083       \ifnum#1=\@ne
2084         \or
2085         \else
2086           ,%
2087         \fi
2088       \the\wd\eql@fieldbox@
2089     }%
2090   \endgroup
2091 }
2092 \def\eql@fieldlength@get#1{%
2093   \ifcase\expandafter#1\eql@fieldlength@tab\fi
2094 }

\eql@tagwidth@get
\eql@tagwidth@save
2095 \let\eql@tagwidth@tab\@empty
2096 \def\eql@tagwidth@get#1{%
2097   \ifcase\expandafter#1\eql@tagwidth@tab\fi
2098 }
2099 \def\eql@tagwidth@save{%

```

```

2100 \begingroup
2101   \let\or\relax
2102   \global\edef\eql@tagwidth@tab{\eql@tagwidth@tab\or\the\eql@tagwidth@}%
2103 \endgroup
2104 }
2105 \def\eql@tagwidth@savezero{%
2106   \begingroup
2107     \let\or\relax
2108     \global\edef\eql@tagwidth@tab{\the\eql@tagwidth@\eql@tagwidth@tab}%
2109   \endgroup
2110 }

```

\eql@colwidth@tab

```
2111 \let\eql@colwidth@tab\@empty
```

columns@colwidth@get

```

2112 \def\eql@columns@colwidth@get#1{%
2113   \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2114 }
2115 \def\eql@columns@colwidth@save{%
2116   \begingroup
2117     \let\or\relax
2118     \global\edef\eql@colwidth@tab{\or\the\wd\thr@\eql@colwidth@tab}%
2119   \endgroup
2120 }

```

J.2 Measure Support

TODO: describe

```

2121 \def\eql@measure@init#1{%
2122   \measuring@true
2123   \eql@row@\z@
2124   \let\displaybreak\eql@displaybreak@measure
2125   \tabskip\z@skip
2126   \everycr{%
2127     \noalign{%
2128       \global\advance\eql@row@\@ne
2129       #1%
2130     }%
2131   }%
2132 }

```

sure@restorecounters

measure@savecounters

```

2133 \let\eql@measure@restorecounters\@empty
2134 \def\eql@measure@savecounters{%
2135   \begingroup
2136     \def\@elt##1{%
2137       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2138     \global\edef\@gtempa{%
2139       \cl@ckpt
2140       \let\noexpand\eql@measure@restorecounters\noexpand\@empty
2141     }%
2142   \endgroup
2143   \let\eql@measure@restorecounters\@gtempa
2144 }

```

J.3 Print Support

TODO: describe

`\eql@print@inithalign`

```
2145 \def\eql@print@init#1{%
2146   \eql@row@z@
2147   \eql@halign@init{%
2148     \global\eql@displaybreak@pen@MM
2149     \global\advance\eql@row@one
2150     #1%
2151   }%
2152 }

2153 \def\eql@print@overfull{%
2154   \dimen@eql@line@width@
2155   \advance\dimen@-hfuzz
2156   \ifdim\dimen@>\displaywidth
2157     \setboxz@hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
2158     \wdz@z@
2159     \htz@\eql@line@height@
2160     \dpz@\eql@line@depth@
2161     \boxz@
2162   \fi
2163 }
```

`\eql@tagbox@print@multi`

```
2164 \def\eql@tagbox@print@multi{%
2165   \advance\eql@tagwidth@-\eql@tagfuzz@
2166   \ifdefined\eql@tagsleft
2167     \eql@display@firstavail@setz@
2168     \ifdim\eql@tagwidth@>\eql@line@avail@
2169       \eql@tagbox@print@tagsleft@raise
2170     \else
2171       \eql@tagbox@print@tagsleft
2172     \fi
2173     \kern\displaywidth
2174   \else
2175     \kern\displaywidth
2176     \advance\eql@tagwidth@\eql@line@width@
2177     \ifdim\eql@tagwidth@>\displaywidth
2178       \eql@tagbox@print@tagsright@raise
2179     \else
2180       \eql@tagbox@print@tagsright
2181     \fi
2182   \fi
2183 }
```

J.4 Line Breaks

TODO: describe

`\eql@math@cr`

```
2184 \protected\def\eql@math@cr{%
```

```

2185 \eql@ampprotecttwo\eql@teststaropt@tight
2186   {\global\eql@displaybreak@pen@=\M\eql@math@cr@}\eql@math@cr@\z@}

```

`\eql@math@cr@`

```

2187 \def\eql@math@cr@[#1]{%
2188   \eql@math@cr@@@
2189   \cr

2190   \noalign{%
2191     \ifnum\eql@displaybreak@pen@=\MM
2192       \penalty\interdisplaylinepenalty
2193     \else
2194       \penalty\eql@displaybreak@pen@
2195     \fi
2196     \advance\eql@vspaceskip@\glueexpr#1\relax%
2197     \vskip\eql@vspaceskip@
2198     \global\eql@vspaceskip@\z@skip
2199   }%
2200 }

```

`\eql@let@cr`

```

2201 \def\eql@let@cr#1{%
2202   \let\\\eql@math@cr
2203   \let\eql@math@cr@@@#1%
2204 }

```

J.5 Intertext

TODO: describe

TODO: revert in everymath?

```

2205 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2206 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2207 \def\eql@intertext@process{%
2208   \eql@math@cr@@@
2209   \cr
2210   \ifmeasuring@
2211     \expandafter\@gobble
2212   \else
2213     \expandafter\eql@intertext@print
2214   \fi
2215 }

```

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

```

2216 \def\eql@intertext@print#1{%
2217   \noalign{%
2218     \eql@halign@after
2219     \let\eql@skip@force@below\z@
2220     \let\eql@skip@force@above\z@
2221     \eql@setkeys{intertext}\eql@intertext@opt
2222     \openup-\eql@spread@
2223     \penalty\postdisplaypenalty

```

```

2224 \ifcase\eq\@skip\@force\@below\relax
2225   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@long\@below\relax
2226 \or
2227   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@short\@below\relax
2228 \or
2229   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@cont\@below\relax
2230 \or
2231   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@par\@below\relax
2232 \or
2233   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@top\@below\relax
2234 \or
2235   \advance\eq\@vspaceskip\@z\@skip
2236 \or
2237   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@med\@below\relax
2238 \or
2239   \advance\eq\@vspaceskip\@glueexpr\eq\@skip\@custom\@below\relax
2240 \fi
2241 \vskip\eq\@vspaceskip\@
2242 \global\eq\@vspaceskip\@z\@skip
2243 \vbox{%
2244   \@parboxrestore
2245   \ifdim
2246     \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2247   \else
2248     \parshape\@ne
2249     \@totalleftmargin\linewidth
2250   \fi
2251   \noindent
2252   \ignorespaces
2253   #1%
2254   \par
2255 }%
2256 \penalty\predisplaypenalty
2257 \ifcase\eq\@skip\@force\@above\relax
2258   \vskip\glueexpr\eq\@skip\@long\@above\relax
2259 \or
2260   \vskip\glueexpr\eq\@skip\@short\@above\relax
2261 \or
2262   \vskip\glueexpr\eq\@skip\@cont\@above\relax
2263 \or
2264   \vskip\glueexpr\eq\@skip\@par\@above\relax
2265 \or
2266   \vskip\glueexpr\eq\@skip\@top\@above\relax
2267 \or
2268   \vskip\z\@skip
2269 \or
2270   \vskip\glueexpr\eq\@skip\@med\@above\relax
2271 \or
2272   \vskip\glueexpr\eq\@skip\@custom\@above\relax
2273 \fi
2274 % \eq\@halign\@prevdepth\@maxdimen
2275 \eq\@halign\@prevdepth\@z\@
2276 \eq\@halign\@before
2277 }
2278 }

```

TODO: describe

```

2279 \newenvironment{eq\@intertext}{%

```

```

2280 \eql@testopt@tight\eql@intertext@{}%
2281 }{%
2282 \aftergroup\eql@intertext@after
2283 \ignorespacesafterend
2284 }

```

TODO: describe

```

2285 \def\eql@intertext@env{intertext}
2286 \def\eql@intertext@[#1]{%
2287 \global\def\eql@intertext@opt{#1}%
2288 \ifx\@currentvir\eql@intertext@env
2289 \expandafter\eql@scan@env\expandafter\eql@intertext@inject
2290 \else
2291 \expandafter\eql@intertext@process
2292 \fi
2293 }

```

TODO: describe

```

2294 \def\eql@intertext@inject{%
2295 \global\edef\eql@intertext@after{%
2296 \noexpand\eql@intertext@process{%
2297 \ifx\eql@scan@body\eql@scan@body@dump
2298 \eql@scan@body@dump
2299 \else
2300 \noexpand\scantokens{\eql@scan@body@dump}%
2301 \fi
2302 }%
2303 }%
2304 }

```

K Equations Box Environment

TODO: describe

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

K.1 Line Breaks

`\eql@box@cr`

```

2305 \protected\def\eql@box@cr{%
2306 \eql@ampprotect\eql@testopt@tight\eql@box@cr@\z@
2307 }
2308 \def\eql@box@cr@[#1]{%
2309 \eql@punct@apply@line
2310 \eql@hook@lineout
2311 \eql@box@lastfield
2312 \cr
2313 \noalign{%
2314 \vskip\glueexpr#1\relax
2315 }%
2316 }

```

K.2 Stacked Mode

```

2317 \def\eq@box@lastfield@lines{&\omit\kern-2\eq@colsep@}
2318 \def\eq@box@open@stacked{%
2319 % \TODO templates
2320 \let\shoveleft\eq@adjust@alignleft
2321 \let\shovecenter\eq@adjust@aligncenter
2322 \let\shoveright\eq@adjust@alignright
2323 \let\eq@box@lastfield\eq@box@lastfield@lines
2324 \eq@halign@init{%
2325 (dev)\eq@dev{starting line \the\eq@row@}%
2326   \global\advance\eq@row@\@ne
2327 }%
2328 \tabskip\z@skip
2329 \halign\bgroup
2330   &%
2331   \eq@shape@pos@\m@ne
2332   \setbox\eq@fieldbox@\hbox{%
2333     \eq@strut@field
2334     \@lign
2335     $\m@th\displaystyle
2336     \eq@hook@colin
2337     ##%
2338     \eq@punct@apply@col
2339     \eq@hook@colout
2340     \eq@tagging@mathsave
2341     $%
2342     \eq@tagging@mathaddlast
2343   }%
2344   \ifnum\eq@shape@pos@=\m@ne
2345     \eq@shape@eval
2346   \fi
2347   \ifcase\eq@shape@pos@
2348     \kern\eq@shape@amount@
2349     \box\eq@fieldbox@
2350     \skip@\@flushglue
2351     \advance\skip@\eq@paddingleft@\relax
2352     \advance\skip@\eq@paddingright@\relax
2353     \advance\skip@-\eq@shape@amount@\relax
2354     \hskip\skip@
2355     \eq@tagging@alignleft
2356   \or
2357     \skip@\@flushglue
2358     \advance\skip@\eq@paddingleft@\relax
2359     \hskip\skip@
2360     \box\eq@fieldbox@
2361     \skip@\@flushglue
2362     \advance\skip@\eq@paddingright@\relax
2363     \hskip\skip@
2364     \eq@tagging@aligncenter
2365   \or
2366     \skip@\@flushglue
2367     \advance\skip@\eq@paddingleft@\relax
2368     \advance\skip@\eq@paddingright@\relax
2369     \hskip\skip@
2370     \box\eq@fieldbox@
2371     \eq@tagging@alignright
2372   \fi
2373   \tabskip\eq@colsep@\relax
2374 \crrc

```

```

2375 \noalign{%
2376 \eql@hook@blockbefore
2377 }%
2378 \eql@hook@blockin
2379 }
2380 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

K.3 Aligned Mode

```

2381 \def\eql@box@lastfield@odd{%
2382 &\omit
2383 \kern-\wd\eql@fieldbox@
2384 \box\eql@fieldbox@
2385 \hfil
2386 &\omit\kern-\eql@colsep@
2387 }%
2388 \def\eql@box@lastfield@even{&\omit\kern-\eql@colsep@}
2389 \def\eql@box@open@aligned{%
2390 % \TODO templates
2391 \let\eql@box@lastfield\@empty
2392 \eql@halign@init{%
2393 (dev)\eql@dev{starting new line}%
2394 }%
2395 \tabskip\z@skip
2396 \halign\bgroup
2397 &%
2398 \let\eql@box@lastfield\eql@box@lastfield@odd
2399 \global\setbox\eql@fieldbox@\hbox{%
2400 \eql@strut@field
2401 \@lign
2402 $\m@th\displaystyle
2403 \eql@hook@colin
2404 ##%
2405 \eql@class@innerleft
2406 \eql@hook@innerleft
2407 \eql@tagging@mathsave
2408 $%
2409 \eql@tagging@mathaddlast
2410 }%
2411 \hfil
2412 \kern\wd\eql@fieldbox@
2413 \tabskip\z@skip
2414 &%
2415 \eql@fieldwidth@\wd\eql@fieldbox@
2416 \kern-\eql@fieldwidth@
2417 \box\eql@fieldbox@
2418 \let\eql@box@lastfield\eql@box@lastfield@even
2419 \llap{\unhbox\eql@fieldbox@}%
2420 \hbox{%
2421 \eql@strut@field
2422 \@lign
2423 $\m@th\displaystyle
2424 \eql@hook@innerright
2425 \eql@class@innerright@sel
2426 ##%
2427 \eql@punct@apply@col
2428 \eql@hook@colout
2429 \eql@tagging@mathsave

```



```

2430     $%
2431     \eql@tagging@mathaddlast
2432   }%
2433   \hfil
2434   \tabskip\eql@colsep@\relax
2435   \crrr
2436   \noalign{%
2437     \eql@hook@blockbefore
2438   }%
2439   \eql@hook@blockin
2440 }
2441 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

K.4 Main

```

2442 \let\eql@box@box\vcenter
2443 \let\eql@box@open\@undefined
2444 \def\eql@box@close{%
2445   \ifvmode\else
2446     \global\eql@totalrows@\eql@row@
2447     \eql@punct@apply@block
2448     \eql@box@cr@[ \z@skip]%
2449   \fi
2450   \crrr
2451   \noalign{%
2452     \eql@hook@blockafter
2453   }%
2454   \eql@tagging@tablesaveinner
2455   \egroup
2456 }

```

\eql@box@start

```

2457 \def\eql@box@start{%
2458   \relax
2459   \ifmmode
2460     \let\eql@box@endmath\@empty
2461   \else
2462     $\let\eql@box@endmath=$%
2463   \fi
2464   \eql@nextopt@process{equationsbox}%
2465   \let\eql@punct@block\eql@punct@main
2466   \let\eql@punct@main\relax
2467   \eql@colsep@\glueexpr\eql@box@colsep\relax
2468   \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
2469   \eql@paddingright@\glueexpr\eql@paddingright@val\relax
2470   \eql@indent@\glueexpr\eql@indent@val\relax
2471   \eql@stack@save@boxed
2472   \let\eql@layoutleft\eql@false
2473   \eql@row@\z@
2474   \eql@totalrows@\@M
2475   \eql@shape@sel
2476   \hskip\glueexpr\eql@box@marginleft\relax
2477   \eql@box@box\bgroup
2478     \eql@display@leave
2479     \let\\\eql@box@cr
2480     \eql@box@open
2481 }

```

`\eql@box@end`

```
2482 \newcommand{\eql@box@end}{%
2483   \eql@box@close
2484   \egroup
2485   \eql@tagging@tableaddinner
2486   \hskip\glueexpr\eql@box@marginright\relax
2487   \eql@stack@restore
2488   \eql@box@endmath
2489 }
```

K.5 Environment

`equationsbox` (*env.*)

```
2490 \newenvironment{equationsbox}{%
2491 (dev)\eql@dev@enterenv
2492   \eql@ampprotect\eql@box@testall\eql@box@start
2493 }{%
2494   \eql@box@end
2495 (dev)\eql@dev@leaveenv
2496 }

2497 \def\eql@box@testall{\eql@box@testtilde}
2498 \def\eql@box@testtilde#1{%
2499   \eql@ifnextgobble@tight~%
2500   {\eqnadopt{lines}\eql@box@testopt{#1}}%
2501   {\eql@box@testopt{#1}}}
2502 \def\eql@box@testopt#1{%
2503   \eql@ifnextchar@tight[%]
2504   {\eql@box@adopt{#1}}%
2505   {#1}}
2506 \def\eql@box@adopt#1[#2]{\eqnadopt{#2}#1}
```

L Single-Line Equation

TODO: describe

L.1 Native Mode

```
2507 \def\eql@single@start@native{%
2508 % \mathopen{}%
2509   \eql@hook@eqin
2510 }%
```

TODO: describe

```
2511 \def\eql@single@end@native{%
2512 % \mathclose{}%
2513   \if@eqnsw
2514     \ifdefined\eql@tagsleft
2515       \leqno
2516     \else
2517       \eqno
2518     \fi
2519   \eql@compose@print
2520   \fi}
```

```

2521 \ifnum\eqldisplaybreakopen@=\@MM\else
2522   \postdisplaypenalty\eqldisplaybreakopen@
2523 \fi
2524 }%

```

L.2 Print

```

2525 \def\eqlsingle@start@print{%
2526   \eql@totalrows@\@ne
2527   \eql@row@\z@
2528   \eql@display@init
2529   \let\shoveleft\eql@adjust@alignleft
2530   \let\shovecenter\eql@adjust@aligncenter
2531   \let\shoveright\eql@adjust@alignright
2532   \eql@adjust@init
2533   \eql@shape@eval
2534   \eql@halign@init{}}%

2535   \eql@row@\@ne
2536   \setbox\eql@fieldbox@\hbox\bgroup
2537     \eql@restore@hfuzz
2538     \eql@strut@field
2539     $\m@th\displaystyle%$
2540     \eql@hook@eqin
2541 }

2542 \def\eqlsingle@end@print{%
2543   \eql@tagging@mathsave
2544   $,$
2545   \hfil
2546   \kern\z@
2547 \egroup

2548 \eql@fieldwidth@\wd\eql@fieldbox@
2549 \eql@line@height@\ht\eql@fieldbox@
2550 \eql@line@depth@\dp\eql@fieldbox@
2551 \eql@totalwidth@\eql@fieldwidth@
2552 \eql@totalrows@\@ne

2553 \if@eqnsw
2554   \eql@tagbox@make\eql@compose@print
2555   \eql@tagrows@\@ne
2556 \else
2557   \eql@tagwidth@\z@
2558   \eql@tagrows@\z@
2559 \fi
2560 \eql@tagwidth@max@\eql@tagwidth@

2561 \eql@adjust@calc

2562 \halign{##\cr
2563   \noalign{\eql@halign@before}}%
2564 \if@eqnsw
2565   \eql@adjust@sel@tag
2566 \else
2567   \eql@adjust@sel@notag
2568 \fi
2569 \cr
2570 \noalign{\eql@halign@after}}%
2571 \eql@tagging@tablesavelines
2572 }%
2573 \eql@row@\tw@
2574 \eql@display@close

```

2575 }

M Multi-Line Lines Mode

M.1 Measure

TODO: describe

```
2576 \def\eq@lines@measure@line@begin{%
2577 <dev>\eq@dev{starting line \the\eq@row}%
2578 \eq@numbering@measure@line@begin
2579 \eq@hook@linein
2580 }
```

TODO: describe

```
2581 \def\eq@lines@measure@line@end{%
2582 \eq@punct@apply@line
2583 \eq@hook@lineout
2584 }
```

TODO: describe

```
2585 \def\eq@lines@measure@field{%
2586 \kern\wd\eq@fieldbox@
2587 }
```

TODO: describe

```
2588 \def\eq@lines@measure@tag{%
2589 \ifnum\eq@numbering@target@<\z@
2590 \if@eqnsw
2591 \eq@tagbox@make\eq@compose@measure
2592 \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2593 \global\eq@tagwidth@max@\eq@tagwidth@
2594 \fi
2595 \global\advance\eq@tagrows@\@ne
2596 \else
2597 \eq@tagwidth@\z@
2598 \fi
2599 \fi
2600 }
```

\eq@lines@measure

```
2601 \def\eq@lines@measure{%
2602 <dev>\eq@dev@center\eq@lines@measure
2603 \eq@tagwidth@max@\z@
2604 \eq@tagrows@\z@
2605 \eq@measure@savecounters
2606 \setbox\z@\vbox{%
2607 \eq@numbering@measure@init
2608 \eq@measure@init\eq@lines@measure@line@begin
2609 \eq@let@cr\eq@lines@measure@line@end
2610 \halign{%
2611 \setbox\eq@fieldbox@\hbox{%
2612 \@lign
2613 $\m@th\displaystyle
2614 \eq@hook@colin
```

```

2615     ##%
2616     \eql@punct@apply@col
2617     \eql@hook@colout
2618     $%
2619     }%
2620     \eql@lines@measure@field
2621     \eql@lines@measure@tag
2622     \crr
2623     \noalign{%
2624     \eql@hook@blockbefore
2625     }%
2626     \eql@hook@blockin
2627     \eql@scan@body
2628     \ifvmode\else
2629     \eql@punct@apply@block
2630     \eql@hook@blockout
2631     \eql@lines@measure@line@end
2632     \cr
2633     \fi
2634     \omit
2635     \cr
2636     \noalign{%
2637     \eql@hook@blockafter
2638     }%
2639     }%
2640     \global\advance\eql@row@-\tw@
2641     \eql@numbering@measure@eval
2642     \ifnum\eql@numbering@target@>\z@
2643     \eql@tagbox@make\eql@compose@measure
2644     \global\eql@tagwidth@max@\eql@tagwidth@
2645     \global\eql@tagrows@\@ne
2646     \fi
2647     }%
2648     \eql@totalrows@\eql@row@
2649     \ifdefined\eql@numbering@subeq@use
2650     \eql@numbering@subeq@test
2651     \fi
2652     \eql@measure@restorecounters
2653     \setbox\z@\vbox{%
2654     \unvbox\z@
2655     \unpenalty
2656     \global\setbox\@ne\lastbox
2657     }%
2658     \eql@totalwidth@\wd\@ne

2659 <dev>\eql@dev@leave\eql@lines@measure
2660 }

```

M.2 Print

TODO: describe

```
mes@print@line@begin
```

```

2661 \def\eql@lines@print@line@begin{%
2662 <dev>\eql@dev{starting line \the\eql@row@}%
2663 \eql@numbering@print@line@begin
2664 \eql@hook@linein

```

2665 }

TODO: describe

```
2666 \def\eq@lines@print@line@end{%
2667   \eq@punct@apply@line
2668   \eq@hook@lineout
2669 }
```

TODO: describe

```
2670 \def\eq@lines@print@line@adjust{%
2671   \eq@numbering@print@line@eval
2672   \eq@fieldwidth@\wd\eq@fieldbox@
2673   \eq@line@height@\ht\eq@fieldbox@
2674   \eq@line@depth@\dp\eq@fieldbox@
2675   \if@eqnsw
2676     \eq@tagbox@make\eq@compose@print
2677     \eq@adjust@sel@tag
2678   \else
2679     \eq@adjust@sel@notag
2680   \fi
2681 }
```

TODO: describe

```
2682 \def\eq@lines@print{%
2683 (dev)\eq@dev@enter\eq@lines@print
2684   \eq@display@init
2685   \eq@adjust@init
2686   \eq@adjust@calc
2687   \eq@numbering@print@init
2688   \eq@print@init\eq@lines@print@line@begin
2689   \eq@let@cr\eq@lines@print@line@end
2690   \tabskip\z@skip
2691   \halign{%
2692     \eq@shape@eval
2693     \setbox\eq@fieldbox@\hbox{%
2694       \eq@restore@hfuzz
2695       \eq@strut@field
2696       \@lign
2697       $\m@th\displaystyle
2698       \eq@hook@colin
2699       ##%
2700       \eq@punct@apply@col
2701       \eq@hook@colout
2702       \eq@tagging@mathsave
2703       $%
2704       \hfil
2705       \kern\z@
2706     }%
2707     \eq@lines@print@line@adjust
2708     \crr
2709     \noalign{%
2710       \eq@halign@before
2711       \eq@numbering@print@block@begin
2712       \eq@hook@blockbefore
2713     }%
2714 % \TODO relax? leavevmode?!
2715   \eq@hook@blockin
2716   \eq@scan@body
```

```

2717 \ifvmode\else
2718 \eql@punct@apply@block
2719 \eql@hook@blockout
2720 \eql@lines@print@line@end
2721 \cr
2722 \fi
2723 \noalign{%
2724 \eql@hook@blockafter
2725 \eql@halign@after
2726 (dev)\eql@dev@leave\eql@lines@print
2727 }%
2728 \eql@tagging@tablesavelines
2729 }%
2730 \eql@display@close
2731 }

```

N Multi-Line Columns Mode

TODO: describe

N.1 Columns Processing

TODO: describe

```

\begin{columns}
\columns@completerow
2732 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
2733 \def\eql@columns@completerow{%
2734 \count@\eql@totalcolumns@
2735 \advance\count@-\eql@column@
2736 \advance\count@\@ne
2737 \edef\eql@tmp{%
2738 \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
2739 \eql@tmp
2740 }

```

N.2 Measure

TODO: describe

```

\begin{measure@line}
2741 \def\eql@columns@measure@line@begin{%
2742 (dev)\eql@dev{starting line \the\eql@row@}%
2743 \global\eql@column@\z@
2744 \eql@numbering@measure@line@begin
2745 \eql@hook@linein
2746 }

2747 \def\eql@columns@measure@field{%
2748 \eql@fieldlength@save\eql@column@
2749 \kern\wd\eql@fieldbox@
2750 }

```

ms@measure@line@end

```
2751 \def\eq@columns@measure@line@end{%
2752   \eq@punct@apply@line
2753   \eq@hook@lineout
2754   &\omit
2755   \ifnum\eq@column@>\eq@totalcolumns@
2756     \global\eq@totalcolumns@\eq@column@
2757   \fi
2758   \eq@columns@measure@tag
2759 }
```

@columns@measure@tag

```
2760 \def\eq@columns@measure@tag{%
2761   \ifnum\eq@numbering@target@<\z@
2762     \if@eqnsw
2763       \eq@tagbox@make\eq@compose@measure
2764       \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2765         \global\eq@tagwidth@max@\eq@tagwidth@
2766       \fi
2767       \global\advance\eq@tagrows@\@ne
2768     \else
2769       \eq@tagwidth@\z@
2770     \fi
2771     \eq@tagwidth@save
2772   \fi
2773 }
```

\eq@columns@measure

```
2774 \def\eq@columns@measure{%
2775 (dev)\eq@dev@enter\eq@columns@measure
2776   \eq@totalcolumns@\z@
2777   \eq@tagwidth@max@\z@
2778   \let\eq@tagwidth@tab\@empty
2779   \let\eq@fieldlength@tab\@empty
2780   \eq@tagrows@\z@
2781   \eq@measure@savecounters
2782   \setbox\z@\vbox{%
2783     \eq@numbering@measure@init
2784     \eq@measure@init\eq@columns@measure@line@begin
2785     \eq@let@cr\eq@columns@measure@line@end
2786     \tabskip\z@skip
2787     \halign{%
2788       &%
2789       \global\advance\eq@column@\@ne
2790       \hfil
2791       \global\setbox\eq@fieldbox@\hbox{%
2792         \@lign
2793         $\m@th\displaystyle
2794         \eq@hook@colin
2795         ##%
2796         \eq@class@innerleft
2797         \eq@hook@innerleft
2798         $%
2799       }%
2800       \global\eq@fieldwidth@\wd\eq@fieldbox@
2801       \eq@columns@measure@field
```



```

2802      &%
2803      \global\advance\eqL@column@\@ne
2804      \setbox\eqL@fieldbox@\hbox{%
2805      \@lign
2806      $\m@th\displaystyle
2807      \eqL@hook@innerright
2808      \eqL@class@innerright@sel
2809      ##%
2810      \eqL@punct@apply@col
2811      \eqL@hook@colout
2812      $%
2813      }%
2814      \eqL@columns@measure@field
2815      \hfil
2816      \crr
2817      \noalign{%
2818      \eqL@hook@blockbefore
2819      }%
2820      \eqL@hook@blockin
2821      \eqL@scan@body

```

TODO: test for vmode okay?!

```

2822      \ifvmode\else
2823      \eqL@punct@apply@block
2824      \eqL@hook@blockout
2825      \eqL@columns@measure@line@end
2826      \cr
2827      \fi
2828      \noalign{%
2829      \eqL@hook@blockafter
2830      }%

```

TODO: should we enforce even columns already here?! **TODO:** should we guard against no columns at all?!

```

2831      \eqL@columns@completerow
2832      \cr
2833      }%
2834      \global\advance\eqL@row@-\tw@
2835      \eqL@numbering@measure@eval
2836      \ifnum\eqL@numbering@target@>\z@
2837      \eqL@tagbox@make\eqL@compose@measure
2838      \global\eqL@tagwidth@max@\eqL@tagwidth@
2839      \global\eqL@tagrows@\@ne
2840      \eqL@tagwidth@savezero
2841      \fi
2842      }%
2843      \eqL@totalrows@\eqL@row@
2844      \ifdefined\eqL@numbering@subeq@use
2845      \eqL@numbering@subeq@test
2846      \fi
2847      \eqL@measure@restorecounters
2848      % \eqL@totalwidth@\wd\z@

2849      \setbox\z@\vbox{%
2850      \unvbox\z@
2851      \unpenalty
2852      \global\setbox\@ne\lastbox
2853      }%

```

```
2854 \eql@totalwidth@wd@ne
```

TODO: why not recycle box contents altogether?!

```
2855 \let\eql@colwidth@tab@empty
2856 \loop
2857   \setbox@ne\hbox{%
2858     \unhbox@ne
2859     \unskip
2860     \global\setbox@thr@@\lastbox
2861   }%
2862 \ifhbox@thr@@
2863   \eql@columns@colwidth@save
2864 \repeat

2865 (dev)\eql@dev@leave\eql@columns@measure
2866 }
```

N.3 Print

TODO: describe

```
mns@print@line@begin
```

```
2867 \def\eql@columns@print@line@begin{%
2868 (dev)\eql@dev{starting line \the\eql@row}%
2869   \global\eql@column@z@
2870   \global\eql@line@pos@\eql@marginleft@
2871   \global\eql@line@width@z@
2872   \global\eql@line@avail@\eql@totalwidth@
2873   \global\eql@line@height@z@
2874   \global\eql@line@depth@z@
2875   \eql@numbering@print@line@begin
2876   \eql@hook@linein
2877 }
```

```
@columns@print@field
```

```
2878 \def\eql@columns@print@field{%
determine available and used space

2879   \dimen@\eql@columns@colwidth@get\eql@column@\relax
2880   \ifdim\wd\eql@fieldbox@>z@
2881     \ifdim\eql@line@width@=z@
2882       \eql@line@avail@\eql@line@pos@
2883       \ifodd\eql@column@
2884         \advance\eql@line@avail@\dimen@
2885         \advance\eql@line@avail@-\wd\eql@fieldbox@
2886       \fi
2887       \global\eql@line@avail@\eql@line@avail@
2888     \fi
2889     \eql@line@width@\eql@line@pos@
2890     \ifodd\eql@column@
2891       \advance\eql@line@width@\dimen@
2892     \else
2893       \advance\eql@line@width@\wd\eql@fieldbox@
2894     \fi
2895     \global\eql@line@width@\eql@line@width@
```

```

2896 \fi
2897 \advance\eq@line@pos@\dimen@
2898 \ifodd\eq@column@\else
2899   \advance\eq@line@pos@\eq@colsep@
2900 \fi
2901 \global\eq@line@pos@\eq@line@pos@

```

update height and depth

```

2902 \ifdim\ht\eq@fieldbox@>\eq@line@height@
2903   \global\eq@line@height@\ht\eq@fieldbox@
2904 \fi
2905 \ifdim\dp\eq@fieldbox@>\eq@line@depth@
2906   \global\eq@line@depth@\dp\eq@fieldbox@
2907 \fi

```

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

```

2908 %
2909 % \box\eq@fieldbox@
2910 %
2911 % \dimen@\eq@columns@colwidth@get\eq@column@\relax
2912 % \advance\dimen@-\wd\eq@fieldbox@
2913 % \ifodd\eq@column@
2914 %   \kern\dimen@
2915 %   \box\eq@fieldbox@
2916 % \else
2917 %   \box\eq@fieldbox@
2918 %   \kern\dimen@
2919 % \fi
2920 %
2921 % \dimen@\eq@columns@colwidth@get\eq@column@\relax
2922 % \ifodd\eq@column@
2923 %   \kern\dimen@
2924 % \else
2925 %   \advance\dimen@-\wd\eq@fieldbox@
2926 %   \box\eq@fieldbox@
2927 %   \kern\dimen@
2928 % \fi
2929 %
2930 }

```

```

2931 \def\eq@columns@print@trailright{%
2932   &\omit
2933   \global\advance\eq@column@\@ne
2934   \setbox\eq@fieldbox@\hbox{%
2935     \kern-\wd\eq@fieldbox@\box\eq@fieldbox@
2936   }%
2937   \eq@columns@print@field
2938 }

```

lums@print@line@end

```

2939 \def\eq@columns@print@line@end{%
2940   \eq@punct@apply@line
2941   \eq@hook@lineout
2942 % \TODO add an even column with empty stuff if box processing deferred
2943   \ifodd\eq@column@
2944     \expandafter\eq@columns@print@trailright

```

```

2945 \fi
2946 \eql@columns@completerow
2947 \eql@columns@print@tag
2948 }

```

`\eql@columns@print@tag`

```

2949 \def\eql@columns@print@tag{%
2950 \dimen@ \eql@totalwidth@
2951 \advance \dimen@ \eql@colsep@
2952 \kern - \dimen@

```

determine first line available space

```

2953 \eql@display@firstavail@set \eql@line@avail@
2954 \eql@print@overfull
2955 \eql@numbering@print@line@eval
2956 \if@eqnsw
2957 \eql@tagbox@make \eql@compose@print
2958 \eql@tagging@tagaddbox
2959 \eql@tagbox@print@multi
2960 \else
2961 \eql@tagging@tagaddbox
2962 \kern \displaywidth
2963 \fi
2964 }

```

`\eql@columns@print`

```

2965 \def\eql@columns@print{%
2966 (dev)\eql@dev@enter \eql@columns@print
2967 \eql@columns@adjust
2968 \eql@display@init
2969 \eql@numbering@print@init
2970 \eql@print@init \eql@columns@print@line@begin
2971 \eql@let@cr \eql@columns@print@line@end
2972 \tabskip \eql@marginleft@
2973 \halign{%
2974 &%
2975 \global \advance \eql@column@ \@ne
2976 \hfil
2977 \global \setbox \eql@fieldbox@ \hbox{%
2978 \eql@strut@field
2979 \@lign
2980 $\m@th \displaystyle
2981 \eql@hook@colin
2982 ##%
2983 \eql@class@innerleft
2984 \eql@hook@innerleft
2985 \eql@tagging@mathsave
2986 $%
2987 \eql@tagging@mathaddlast
2988 }%
2989 \global \eql@fieldwidth@ \wd \eql@fieldbox@
2990 \eql@columns@print@field
2991 \tabskip \z@skip
2992 &%
2993 \global \advance \eql@column@ \@ne
2994 \setbox \eql@fieldbox@ \hbox{%

```

```

2995 % \TODO printing left field in right field
2996     \kern-\wd\eq\fieldbox@
2997     \box\eq\fieldbox@
2998     \eq\strut@field
2999     \@lign
3000     $\m@th\displaystyle
3001     \eq\hook@innerright
3002     \eq@class@innerright@sel
3003     ##%
3004     \eq@punct@apply@col
3005     \eq\hook@colout
3006     \eq>tagging@mathsave
3007     $%
3008     \eq>tagging@mathaddlast
3009     }%
3010     \eq@columns@print@field
3011     \hfil
3012     \tabskip\eq@colsep@\relax
3013     \crrc
3014     \noalign{%
3015         \eq@halign@before
3016         \eq@numbering@print@block@begin
3017         \eq\hook@blockbefore
3018     }%
3019     \eq\hook@blockin
3020     \eq@scan@body
3021     \ifvmode\else
3022         \eq@punct@apply@block
3023         \eq\hook@blockout
3024         \eq@columns@print@line@end
3025     \cr
3026     \fi
3027     \noalign{%
3028         \eq\hook@blockafter
3029         \eq@halign@after
3030 (dev)\eq@dev@leave\eq@columns@print
3031     }%
3032     \eq>tagging@tablesavealign
3033     }%
3034     \eq@display@close
3035 }

```

N.4 Adjust

TODO: describe **TODO:** does this respect the margin for numbers in centre mode?

```

3036 \def\eq@columns@adjust{%
3037     \eq@colsepmin@\glueexpr\eq@colsepmin@val\relax

```

TODO: shouldn't we do this earlier for access to last column?

```

3038     \ifodd\eq@totalcolumns@
3039         \advance\eq@totalcolumns@\@ne
3040     \fi

```

TODO: should we guard against no columns?!

```

3041     \ifnum\eq@totalcolumns@<\thr@@

```

```

3042 \let\eql@columns@margins\eql@true
3043 \fi

```

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

```

3044 \eql@columns@inter@\eql@totalcolumns@
3045 \divide\eql@columns@inter@\tw@
3046 \advance\eql@columns@inter@\m@ne

3047 \ifdefined\eql@layoutleft
3048 \eql@layoutleftmargin@\glueexpr\eql@layoutleftmargin@val\relax
3049 \else
3050 \eql@adjust@tagmargin
3051 \fi

3052 \eql@colsep@\displaywidth
3053 \advance\eql@colsep@-\eql@totalwidth@
3054 \ifdefined\eql@layoutleft
3055 \advance\eql@colsep@-\eql@layoutleftmargin@
3056 \else
3057 \advance\eql@colsep@-\eql@tagmargin@
3058 \ifdefined\eql@columns@margins\else
3059 \ifdim\eql@tagmargin@>\z@
3060 \advance\eql@colsep@-\eql@tagsepmin@
3061 \fi
3062 \fi
3063 \fi
3064 \count@\eql@columns@inter@
3065 \ifdefined\eql@columns@margins
3066 \ifdefined\eql@layoutleft
3067 \advance\count@\@ne
3068 \else
3069 \advance\count@\tw@
3070 \fi
3071 \fi
3072 \divide\eql@colsep@\count@

```

TODO: here or above, this code does not make much sense if there is a single column. nevertheless it works using the following code. yet it could be cleaner to treat a single column separately (may be some distinctions based on flush left)

```

3073 \ifdim\eql@colsep@<\eql@colsepmin@
3074 \eql@colsep@\eql@colsepmin@
3075 \else
3076 \dimen@\glueexpr\eql@colsepmax@val\relax
3077 \ifdim\eql@colsep@>\dimen@
3078 \eql@colsep@\dimen@
3079 \fi
3080 \fi

```

Now determine the left margin `\eql@marginleft@` **TODO:** complete

```

3081 \ifdefined\eql@layoutleft
3082 \ifdim\eql@colsep@=\eql@colsepmin@
3083 \eql@marginleft@\displaywidth
3084 \advance\eql@marginleft@-\eql@totalwidth@
3085 \advance\eql@marginleft@-\eql@columns@inter@\eql@colsep@
3086 \ifdim\eql@marginleft@>\eql@layoutleftmargin@
3087 \eql@marginleft@\eql@layoutleftmargin@
3088 \else

```

```

3089     \ifdim\eql@marginleft@<\eql@layoutleftmarginmin@
3090         \eql@marginleft@\eql@layoutleftmarginmin@
3091     \fi
3092 \fi
3093 \else
3094     \eql@marginleft@\eql@layoutleftmargin@
3095 \fi
3096 \else
3097     \ifdefined\eql@columns@margins
3098         \eql@marginleft@\displaywidth
3099         \advance\eql@marginleft@-\eql@totalwidth@
3100         \advance\eql@marginleft@-\eql@columns@inter@\eql@colsep@
3101         \ifdim\eql@marginleft@<\eql@tagmargin@
3102             \eql@marginleft@\z@
3103         \else
3104             \advance\eql@marginleft@-\eql@tagmargin@
3105             \divide\eql@marginleft@\tw@
3106         \fi
3107         \ifdefined\eql@tagsleft
3108             \advance\eql@marginleft@\eql@tagmargin@
3109         \fi
3110     \else
3111         \ifdefined\eql@tagsleft
3112             \eql@marginleft@\eql@tagmargin@
3113             \ifdim\eql@tagmargin@>\z@
3114                 \advance\eql@marginleft@\eql@tagsepmin@
3115             \fi
3116         \else
3117             \eql@marginleft@\z@
3118         \fi
3119     \fi
3120 \fi

```

Loop through the rows and adjust the intercolumn and margin space to make the tags fit into the available space at the corresponding side as far as possible:

```

3121 \ifdefined\eql@tagsleft
3122     \let\eql@columns@adjust@test\eql@columns@adjust@test@tagleft
3123 \else
3124     \let\eql@columns@adjust@test\eql@columns@adjust@test@tagright
3125 \fi
3126 \eql@row@\eql@totalrows@
3127 \loop\ifnum\eql@row@>\z@

```

Fetch the tag width for the current row depending on whether there are tags for individual rows or one overall tag. If a tag is present, compute the available space and try to adjust spaces if needed:

```

3128     \ifnum\eql@numbering@target@<\z@
3129         \eql@tagwidth@\eql@tagwidth@get\eql@row@\relax
3130     \else
3131         \ifnum\eql@numbering@target@=\eql@row@
3132             \eql@tagwidth@\eql@tagwidth@get\z@\relax
3133         \fi
3134     \fi
3135     \ifdim\eql@tagwidth@>\z@
3136         \eql@columns@adjust@calc
3137         \eql@columns@adjust@test
3138     \fi
3139     \advance\eql@row@\m@ne

```

```
3140 \repeat
```

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

```
3141 \advance\eql@totalwidth@\eql@columns@inter@\eql@colsep@
3142 \advance\eql@totalwidth@\eql@marginleft@
3143 }
```

Calc Space.

`\eql@columns@adjust@calc` Compute the space that is available at the beginning and at the end of a particular row `\eql@row@`. The space available at the beginning is returned in `\eql@line@avail@` and `\@tempcnta` counts the available intercolumn spaces whose width is not contained in `\eql@line@avail@` because it is still flexible at this stage. The total used width is returned in `\eql@line@width@` and `\@tempcntb` describes the last used intercolumn space. The available space at the end of the row is given as the difference to `\eql@totalwidth@` and `\eql@columns@inter@`:

```
3144 \def\eql@columns@adjust@calc{%
3145 \eql@line@pos@ \z@
3146 \eql@column@ \z@
3147 \eql@line@avail@\eql@totalwidth@
3148 \@tempcnta\eql@totalcolumns@
3149 \eql@line@width@\z@
3150 \@tempcntb \z@
3151 \edef\@tempb{\eql@fieldlength@get\eql@row@}%
3152 \@for\@tempa:=\@tempb\do
3153 \eql@columns@adjust@calc@col
3154 \advance\@tempcnta\m@ne
3155 \divide\@tempcnta\tw@
3156 \advance\@tempcntb\m@ne
3157 \divide\@tempcntb\tw@
3158 }
```

`\eql@columns@adjust@calc@col` The macro `\eql@columns@adjust@width@col` iterates over columns. When a non-blank field 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:

```
3159 \def\eql@columns@adjust@calc@col{%
3160 \advance\eql@column@\@ne
3161 \@tempdima\@tempa\relax
3162 \dimen@eql@columns@colwidth@get\eql@column@\relax
3163 \ifdim\@tempdima>\z@
3164 \ifdim\eql@line@width@=\z@
3165 \eql@line@avail@\eql@line@pos@
3166 \@tempcnta\eql@column@
3167 \ifodd\eql@column@
3168 \advance\eql@line@avail@\dimen@
3169 \advance\eql@line@avail@-\@tempdima
3170 \fi
3171 \fi
3172 \eql@line@width@\eql@line@pos@
3173 \@tempcntb\eql@column@
3174 \ifodd\eql@column@
3175 \advance\eql@line@width@\dimen@
3176 \else
3177 \advance\eql@line@width@\@tempdima
```



```

3178   \fi
3179   \fi
3180   \advance\eql@line@pos@\dimen@
3181 }

```

Placement for Right Tags.

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

```
3182 \def\eql@columns@adjust@test@tagright{%
```

The register `\@tempdima` will hold the amount of available space.

```

3183   \@tempdima\displaywidth
3184   \advance\@tempdima-\eql@line@width@
3185   \advance\@tempdima-\eql@tagwidth@

```

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

```

3186   \dimen@\eql@marginleft@
3187   \advance\dimen@\@tempcntb\eql@colsep@
3188   \ifdim\dimen@>\@tempdima

```

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).

```

3189   \dimen@\@tempcntb\eql@colsepmin@
3190   \ifdefined\eql@layoutleft
3191     \advance\dimen@\eql@layoutleftmarginmin@
3192     \fi
3193   \ifdim\dimen@>\@tempdima\else

```

If so, hand over to `\eql@columns@adjust@modify@tagright`.

```

3194     \eql@columns@adjust@modify@tagright
3195     \fi
3196   \fi
3197 }

```

`just@modify@tagright` Adjust the intercolumn space and left margin to make the row fit.

```
3198 \def\eql@columns@adjust@modify@tagright{%
```

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.

```

3199   \ifnum\@tempcntb>\z@
3200     \dimen@\@tempdima
3201     \count@\@tempcntb
3202     \ifdefined\eql@layoutleft
3203       \advance\dimen@-\eql@marginleft@
3204     \else
3205       \ifdefined\eql@columns@margins
3206         \advance\count@\@ne
3207       \fi
3208     \fi
3209     \divide\dimen@\count@

```

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

```

3210   \ifdim\dimen@<\eql@colsep@
3211     \ifdim\dimen@<\eql@colsepmin@
3212       \dimen@\eql@colsepmin@
3213     \fi
3214   \eql@colsep@\dimen@
3215 \fi
3216 \fi

```

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

```

3217 \dimen@\@tempdima
3218 \advance\dimen@-\@tempcntb\eql@colsep@
3219 \ifdim\dimen@<\eql@marginleft@
3220   \eql@marginleft@\dimen@
3221 \fi
3222 }

```

Placement for Left Tags.

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

```

3223 \def\eql@columns@adjust@test@tagleft{%

```

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.

```

3224 \count@\eql@columns@inter@
3225 \advance\count@-\@tempcnta
3226 \@tempdima\eql@tagwidth@
3227 \advance\@tempdima-\eql@line@avail@

```

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

```

3228 \dimen@\eql@marginleft@
3229 \advance\dimen@\@tempcnta\eql@colsep@
3230 \ifdim\dimen@<\@tempdima

```

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

```

3231   \ifdefined\eql@layoutleft
3232     \dimen@\eql@layoutleftmarginmax@
3233   \else
3234     \dimen@\displaywidth
3235   \fi
3236 \ifdim\dimen@>\eql@tagwidth@

```

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

```

3237   \dimen@\count@\eql@colsepmin@
3238   \advance\dimen@\eql@totalwidth@
3239   \advance\dimen@\@tempdima
3240   \ifdim\dimen@>\displaywidth\else

```

If so, hand over to `\eql@columns@adjust@modify@tagleft`.

```

3241         \eql@columns@adjust@modify@tagleft
3242     \fi
3243 \fi
3244 \fi
3245 }

```

TODO: implement a maximum shift (if tag+sep exceeds max, don't adjust) **TODO:** could this mechanism possibly shift any longer line past the margin?!

`djust@modify@tagleft` Adjust the intercolumn space and left margin to make the row fit.

```

3246 \def\eql@columns@adjust@modify@tagleft{%

```

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.

```

3247 \ifnum\count@>\z@
3248     \dimen@\displaywidth
3249     \advance\dimen@-\eql@totalwidth@
3250     \advance\dimen@-\@tempdima
3251     \ifdefined\eql@columns@margins
3252         \advance\count@\@ne
3253     \fi
3254     \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.

```

3255     \ifdim\dimen@<\eql@colsep@
3256         \ifdim\dimen@<\eql@colsepmin@
3257             \dimen@\eql@colsepmin@
3258         \fi
3259         \advance\dimen@-\eql@colsep@
3260         \advance\eql@marginleft@-\eql@columns@inter@\dimen@
3261         \advance\eql@colsep@\dimen@
3262     \fi
3263 \fi

```

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

```

3264 \dimen@-\@tempcnta\eql@colsep@
3265 \advance\dimen@\@tempdima
3266 \ifdim\dimen@>\eql@marginleft@
3267     \eql@marginleft@\dimen@
3268 \fi
3269 }

```

O Interface

O.1 Scanning the Equation Body

The multi-line equation environment must scan its body twice: once to determine how wide the columns are and then to actually typeset them. This means that we must collect all text in this body before calling the environment macros. The mechanism and its description follows `amsmath` closely.

Token Register.

`\eql@scan@reg@` We start by defining a token register to hold the equation body.

```
3270 \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:

```
3271 \def\eql@scan@body@dump{\the\eql@scan@reg@}  
3272 \def\eql@scan@body@rescan{%  
3273   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}  
3274 \let\eql@scan@body\eql@scan@body@dump
```

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

```
3275 \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@`:

```
3276 \def\eql@scan@env#1{%  
3277   (dev)\eql@dev@enter\eql@scan@env  
3278   \def\eql@scan@end{#1}\expandafter\end\expandafter{\@currenenv}}%  
3279   \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`:

```
3280 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenenv\endcsname}%  
3281 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate  
3282 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate  
3283 \eql@scan@iterate  
3284 }
```

`\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:

```
3285 \def\eql@scan@env@iterate#1\end#2{%  
3286   \edef\eql@scan@stack{%  
3287     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%  
3288   \ifx\@empty\eql@scan@stack  
3289     \@checkend{#2}%  
3290     \eql@scan@addto{#1}%
```

```

3291 \expandafter\let\eql@scan@iterate\eql@scan@env@org
3292 (dev)\eql@dev@leave\eql@scan@env
3293 \expandafter\eql@scan@end
3294 \else
3295 \eql@scan@addto{#1\end{#2}}%
3296 \expandafter\eql@scan@iterate
3297 \fi
3298 }

```

`\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.

```

3299 \long\def\eql@scan@env@count#1\begin#2{%
3300 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
3301 }

```

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

```

3302 \def\eql@scan@env@cancel{%
3303 \@namedef{end\@currenenvir}{\ignorespacesafterend}%
3304 }

```

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 `\]`:

```

3305 \def\eql@scan@sqr#1{%
3306 (dev)\eql@dev@enter\eql@scan@sqr
3307 \def\eql@scan@end{#1\]}%
3308 \eql@scan@reg@{\def\eql@scan@stack{b}%
3309 \let\eql@scan@sqr@org\[%\]
3310 \let\[\eql@scan@sqr@iterate%\]
3311 \[%\]
3312 }

```

Iterate until we find a balanced pairing of square brackets. Then call the call-back macro:

```

3313 \def\eql@scan@sqr@iterate#1\]%
3314 \edef\eql@scan@stack{%
3315 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
3316 \ifx\@empty\eql@scan@stack
3317 \let\[\eql@scan@sqr@org%\]
3318 \eql@scan@addto{#1}%
3319 (dev)\eql@dev@leave\eql@scan@sqr
3320 \expandafter\eql@scan@end
3321 \else
3322 \eql@scan@addto{#1\]}%
3323 \expandafter\[%\]
3324 \fi
3325 }

```

Push a 'b' for every encountered instance of `\[`:

```

3326 \long\def\eql@scan@sqr@count#1\[#2%\]
3327 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
3328 }

```

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

```
3329 \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 `\>`:

```
3330 \def\eql@scan@ang#1{%
3331 (dev)\eql@dev@enter\eql@scan@ang
3332 \def\eql@scan@end{#1}>}%
3333 \eql@scan@reg@{}\def\eql@scan@stack{b}%
3334 \let\eql@scan@ang@org\<%\>
3335 \let\<\eql@scan@ang@iterate%\>
3336 \<%\>
3337 }
```

Iterate until we find a balanced pairing of angle brackets:

```
3338 \def\eql@scan@ang@iterate#1\>{%
3339 \edef\eql@scan@stack{%
3340 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
3341 \ifx\@empty\eql@scan@stack
3342 \let\<\eql@scan@ang@org%\>
3343 \eql@scan@addto{#1}%
3344 (dev)\eql@dev@leave\eql@scan@ang
3345 \expandafter\eql@scan@end
3346 \else
3347 \eql@scan@addto{#1}>}%
3348 \expandafter\<%\>
3349 \fi
3350 }
```

Push a ‘b’ for every encountered instance of ‘<’:

```
3351 \long\def\eql@scan@ang@count#1\<#2{%\>
3352 \ifx>#2\else b\expandafter\eql@scan@ang@count\fi
3353 }
```

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 `\eqnadopt`. 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:

```
3354 \def\eql@equations@testall{\eql@equations@testtilde}
3355 \def\eql@equations@testtilde#1{%
3356 \eql@ifnextgobble@tight~%
3357 {\eqnadopt{lines}\eql@equations@testopt{#1}}%
3358 {\eql@equations@testopt{#1}}}
3359 \def\eql@equations@testopt#1{%
3360 \eql@ifnextchar@tight[%]
3361 {\eql@equations@adopt{\eql@equations@testexcl{#1}}}%
3362 {\eql@equations@testexcl{#1}}}
3363 \def\eql@equations@adopt#1[#2]{\eqnadopt{#2}#1}
```

```

3364 \def\eqlequations@testexcl#1{%
3365   \eql@ifnextgobble@tight!%
3366   {\eqnadopt{donumber}\eqlequations@testat{#1}}%
3367   {\eqlequations@teststar{#1}}}
3368 \def\eqlequations@teststar#1{%
3369   \eql@ifstar@tight%
3370   {\eqnadopt{nonumber}\eqlequations@testat{#1}}%
3371   {\eqlequations@testat{#1}}}
3372 \def\eqlequations@testat#1{%
3373   \eql@ifat@tight
3374   {\eqlequations@addlabel{#1}}%
3375   {#1}}
3376 \def\eqlequations@addlabel#1#2{\eqnadopt{label={#2}}#1}

```

`equations@processopt` The macro `\eqlequations@processopt` processes the options received by `\eqnadopt`. 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:

```

3377 \def\eqlequations@processopt{%
3378   \let\eql@blocklabel\@undefined
3379   \let\eql@blocktag\@undefined
3380   \let\eql@skip@force@above\@undefined
3381   \let\eql@skip@force@below\@undefined
3382   \let\eql@skip@force@leave\@undefined
3383   \eql@abovespace@z@skip
3384   \eql@belowspace@z@skip
3385   \eql@displaybreak@prepen@\@MM
3386   \eql@nextopt@process{equations}%
3387   \let\eql@punct@block\eql@punct@main
3388   \let\eql@punct@main\relax
3389   \eql@indent@\glueexpr\eql@indent@val\relax
3390   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
3391 }

```

O.3 Single-Line Main

```

3392 \def\eql@single@cr{%
3393   \eql@error{Cannot use ‘\string\’ within display equation.
3394   Please switch to equations environment}%
3395 }

```

TODO: describe

```

3396 \def\eql@single@start{%
3397   \eql@halign@catchprevdepth
3398   \eql@tagging@start
3399   \eql@dollar@begin
3400   \eql@numbering@eval@mode
3401   \let\eql@numbering@subeq@use\eql@false
3402   \eql@stack@save@single

```

TODO: make other display environments push these!?

```

3403   \eql@numbering@single@init
3404   \ifdefined\eql@single@native
3405     \let\eql@single@start@sel\eql@single@start@native
3406     \let\eql@single@end@sel\eql@single@end@native
3407     \let\raisetag\eql@raisetag@default

```

```

3408 \else
3409   \let\eql@single@start@sel\eql@single@start@print
3410   \let\eql@single@end@sel\eql@single@end@print
3411 \fi
3412 \ifdefined\eql@single@crerror\else
3413   \let\\eql@single@cr
3414 \fi
3415 \eql@single@start@sel
3416 }

3417 \def\eql@single@end{%
3418   \eql@punct@apply@block
3419   \eql@hook@eqout
3420   \eql@single@end@sel
3421   \eql@stack@restore
3422   \eql@dollar@dollar@end
3423   \eql@tagging@end
3424   \eql@halign@leave
3425 }

```

TODO: : try to feed in tagging after catchprevdepth

```

3426 \def\eql@single@main{%
3427   \expandafter\eql@single@start
3428   \eql@scan@body
3429   \eql@single@end
3430 }

```

TODO: describe

```

3431 \def\eql@mode@equation{%
3432   \ifdefined\eql@single@doscan
3433     \let\eql@equations@main\eql@single@main
3434     \let\eql@equations@end\@empty
3435   \else
3436     \let\eql@equations@main\@undefined
3437     \let\eql@equations@end\eql@single@end
3438   \fi
3439 }

```

O.4 Multi-Line Main

TODO: note that switching from align to lines mode, the width can be incorrect due to different formatting (punctuation only?!). only minor discrepancies expected and lines can adjust

`\eql@multi@main`

```

3440 \let\eql@multi@mode@lines\eql@false
3441 \def\eql@multi@main{%
3442   \eql@halign@catchprevdepth
3443   \eql@tagging@start
3444   \eql@dollar@dollar@begin
3445   \eql@numbering@eval@mode
3446   \eql@stack@save@multi
3447   \ifdefined\eql@subequations@active
3448     \let\eql@numbering@subeq@use\eql@false
3449   \fi
3450   \ifdefined\eql@numbering@subeq@use
3451     \eql@numbering@subeq@init
3452   \fi

```



```

3453 \let\intertext\eql@intertext
3454 \let\endintertext\endeql@intertext
3455 \let\shoveleft\eql@adjust@alignleft
3456 \let\shovecenter\eql@adjust@aligncenter
3457 \let\shoveright\eql@adjust@alignright
3458 \ifdefined\eql@multi@mode@lines
3459   \expandafter\eql@lines@measure
3460 \else
3461   \ifdefined\eql@ampproof@active
3462     \eql@ampproof
3463   \fi
3464   \expandafter\eql@columns@measure
3465 \fi
3466 \ifx\eql@numbering@subeq@use\@ne
3467   \eql@numbering@subeq@revert
3468 \fi
3469 \ifdefined\eql@multi@mode@lines\else
3470   \ifdefined\eql@multi@linesfallback
3471     \ifnum\eql@totalcolumns@=\@ne
3472       \let\eql@multi@mode@lines\eql@true
3473 %       \eql@lines@measure
3474     \fi
3475   \fi
3476 \fi
3477 \ifdefined\eql@multi@mode@lines
3478   \expandafter\eql@lines@print
3479 \else
3480   \expandafter\eql@columns@print
3481 \fi
3482 \ifdefined\eql@numbering@subeq@use
3483   \eql@numbering@subeq@close
3484 \fi
3485 \eql@stack@restore
3486 \eql@dollar@end
3487 \eql@tagging@end
3488 \eql@halign@leave
3489 }

```

TODO: describe

```

3490 \def\eql@mode@columns{%
3491   \let\eql@equations@main\eql@multi@main
3492   \let\eql@equations@end\@empty
3493   \let\eql@multi@mode@lines\eql@false
3494 }
3495 \def\eql@mode@lines{%
3496   \let\eql@equations@main\eql@multi@main
3497   \let\eql@equations@end\@empty
3498   \let\eql@multi@mode@lines\eql@true
3499 }

```

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 `\eql@equations@start`:

```

3500 \newenvironment{equations}{%
3501 (dev)\eql@dev@enterenv
3502 \ifmmode
3503   \eql@error@mathmode{\string\begin{\@currenvir}}%
3504   \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
3505 \else
3506   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3507   \expandafter\eql@equations@start
3508 \fi
3509 }{%
3510 \eql@equations@end
3511 \ignorespacesafterend
3512 (dev)\eql@dev@leaveenv
3513 }
```

`\eql@equations@start` The macro `\eql@equations@start` first processes the arguments. Depending on the chosen mode of operation, scan the environment body passing on to `\eql@equations@main` or process a single-line equation via `\eql@single@start`:

```

3514 \def\eql@equations@start{%
3515 \eql@equations@processopt
3516 \ifdefined\eql@equations@main
3517   \expandafter\eql@scan@env\expandafter\eql@equations@main
3518 \else
3519   \expandafter\eql@single@start
3520 \fi
3521 }
```

Square Brackets.

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

```

3522 \newenvironment{equations@sqr}{}{}
```

`\eql@equations@sqr@open` The macro `\eql@equations@sqr@open` holds the definition for `\[`. If already in math mode, ignore the enclosed contents. Otherwise add the default arguments `\eql@equations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eql@equations@sqr@start`:

```

3523 \protected\def\eql@equations@sqr@open{%
3524 \ifmmode
3525   \eql@error@mathmode{\string\[...\string\]}%
3526   \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
3527 \else
3528 (dev)\eql@dev@enter{\string\[...\string\]}%
3529   \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
3530   \begin{equations@sqr}%
3531   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3532   \expandafter\eql@equations@sqr@start
3533 \fi
3534 }
```

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

```

3535 \def\eqlequations@sqr@start{%
3536   \eqlequations@processopt
3537   \ifdefined\eqlequations@main
3538     \expandafter\eql@scan@sqr\expandafter\eqlequations@main
3539   \else
3540     \expandafter\eql@single@start
3541   \fi
3542 }

```

`\eqlequations@sqr@close` The macro `\eqlequations@sqr@close` holds the definition for ‘\’.

```

3543 \protected\def\eqlequations@sqr@close{%
3544   \eqlequations@end
3545 (dev)\eql@dev@leave{\[...\string\]}%
3546   \end{equations@sqr}%
3547   \ignorespaces
3548 }

```

Angle Brackets.

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

```

3549 \newenvironment{equations@ang}{\{}}{\}

```

`\eqlequations@ang@open` The macro `\eqlequations@ang@open` holds the definition for ‘\<’.

```

3550 \protected\def\eqlequations@ang@open{%
3551   \ifmmode
3552     \eql@error@mathmode{\string\<...\string\>}%
3553     \expandafter\eql@scan@ang\expandafter\eql@scan@sqrang@cancel
3554   \else
3555 (dev)\eql@dev@enter{\<...\string\>}%
3556     \expandafter\eqnaddopt\expandafter{\eqlequations@ang@opt}%
3557     \begin{equations@ang}%
3558     \expandafter\eql@amprotect\expandafter\eqlequations@testall
3559     \expandafter\eqlequations@ang@start
3560   \fi
3561 }

```

`\eqlequations@ang@start` Process arguments and start handling the equation:

```

3562 \def\eqlequations@ang@start{%
3563   \eqlequations@processopt
3564   \ifdefined\eqlequations@main
3565     \expandafter\eql@scan@ang\expandafter\eqlequations@main
3566   \else
3567     \expandafter\eql@single@start
3568   \fi
3569 }

```

`\eqlequations@ang@close` The macro `\eqlequations@ang@close` holds the definition for ‘\>’.

```

3570 \protected\def\eqlequations@ang@close{%
3571   \eqlequations@end
3572 (dev)\eql@dev@leave{\<...\string\>}%
3573   \end{equations@ang}%
3574   \ignorespaces
3575 }

```

P Options

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

```
3576 \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

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

If no corresponding value is found in the list, an error message is invoked. Single expansion is applied to the list of values:

```
3577 \def\eql@decide@select#1#2#3#4{%
3578   \def\@tempa{#3}%
3579   \let\@tempd\undefined
3580   \@for\@tempc:=#4\do{%
3581     \ifdefined\@tempd\else
3582       \edef\@tempb{\noexpand\@tempb:=\expandafter\@firstoftwo\@tempc}%
3583       \expandafter\@for\@tempb\do{%
3584         \ifx\@tempa\@tempb
3585           \expandafter\expandafter\expandafter\def
3586           \expandafter\expandafter\expandafter\@tempd
3587           \expandafter\expandafter\expandafter{%
3588             \expandafter\@secondoftwo\@tempc}%
3589         \fi
3590       }%
3591     \fi
3592   }%
3593   \ifdefined\@tempd
3594     \@tempd
3595   \else
3596     \eql@error{undefined value ‘#3’ for option ‘#2’ of ‘#1’}%
3597   \fi
3598 }
```

`\eql@decide@if` We will often have to decide between `true` and `false` or related pairs of values:

```
3599 \def\eql@decide@if#1#2#3#4#5{%
3600   \eql@decide@select{#1}{#2}{#3}{%
3601     {on,true,yes,enabled,1}{#4}},%
3602     {{off,false,no,disabled,0}{#5}}}
```

`\eql@decide@bool` Boolean values frequently need to be stored into conditional registers:

```
3603 \def\eql@decide@bool#1#2#3#4{%
3604   \eql@decide@if{#1}{#2}{#3}{\let#4\eql@true}{\let#4\eql@false}}
```

`\eql@decide@abovebelow` **TODO:** describe

```
3605 \def\eql@decide@abovebelow#1#2#3#4#5{%
3606   \eql@decide@select{#1}{#2}{#3}{%
```

```

3607    {,abovebelow,both,tb}{#4#5},%
3608    {above,top,t}{#4},%
3609    {below,bottom,b}{#5}}

```

`\eql@decide@situation` **TODO:** describe

```

3610 \def\eql@decide@situation#1#2#3#4{%
3611   \eql@decide@select{#1}{#2}{#3}{%
3612     {{long}}{\def#4{0}}},%
3613     {{short}}{\def#4{1}}},%
3614     {{cont}}{\def#4{2}}},%
3615     {{par}}{\def#4{3}}},%
3616     {{top}}{\def#4{4}}},%
3617     {{noskip}}{\def#4{5}}},%
3618     {{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 it is hidden from the user in error messages:

```

3619 \def\eql@define@key#1#2{%
3620   \eql@ifnextchar@loose[%
3621     {\eql@definekey@opt{#1}{#2}}%
3622     {\eql@definekey@noopt{#1}{#2}}%
3623   }
3624 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{{#3}}}
3625 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{{#3}{#4}}}
3626 \def\eql@definekey@for#1#2#3{%
3627   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}{#3}}%
3628   \edef\eql@for@vara{\noexpand\eql@for@vara:=#1}%
3629   \expandafter\@for\eql@for@vara\do{%
3630     \edef\eql@for@varb{\noexpand\eql@for@varb:=#2}%
3631     \expandafter\@for\eql@for@varb\do{%
3632       \edef\eql@for@call##1{%
3633         \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@vara}}%
3634       \eql@for@call{##1}}%
3635     }%
3636   }%
3637 }

```

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

```

3638 \def\eql@setkeys#1#2{%
3639   \def\eql@tmp{\setkeys{eql@#1}}%
3640   \expandafter\eql@tmp\expandafter{#2}%
3641 }

```

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

```

\eql@nextopt
  \eqnaddopt
3642 \let\eql@nextopt\@empty
3643 \def\eql@nextopt@process#1{%
3644   (dev)\eql@dev@start\eql@nextopt@process
3645   \eql@setkeys{#1}\eql@nextopt
3646   \let\eql@tagging@opt\eql@nextopt

```

```

3647 \global\let\eql@nextopt\@empty
3648 }
3649 \newcommand{\eqnaddopt}[1]{%
3650 \expandafter\def\expandafter\eql@nextopt\expandafter{\eql@nextopt,#1}}

```

P.3 Options Declarations

TODO: describe

Modes for Equations Box Environment. **TODO:** describe

```

3651 \eql@define@key{equationsbox}{gathered,gather,ga,lines,ln,\string~}[]{%
3652 \eql@mode@stacked}
3653 \eql@define@key{equationsbox}{aligned,align,al,columns,col,@}[]{%
3654 \eql@mode@aligned}
3655 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
3656 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
3657 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
3658 \eql@define@key{equationsbox}{colsep}{\def\eql@box@colsep{#1}}

```

Modes for Equations Environment. **TODO:** describe

```

3659 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@equation}
3660 \eql@define@key{equations}{gathered,gather,ga,lines,ln,\string~}[]{%
3661 \eql@mode@lines}
3662 \eql@define@key{equations}{aligned,align,al,columns,col,@}[]{%
3663 \eql@mode@columns}
3664 \eql@define@key{equations}{native}[true]{%
3665 \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
3666 \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
3667 \eql@define@key{setup}{native}[true]{%
3668 \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
3669 \eql@define@key{setup}{scanequation}[true]{%
3670 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
3671 \eql@define@key{setup}{sqropt}[]{%
3672 \def\eql@equations@sqr@opt{equation,#1}}
3673 \eql@define@key{setup}{angopt}[]{%
3674 \def\eql@equations@ang@opt{align,#1}}

```

Vertical Spacing. **TODO:** set at end of env only! **TODO:** describe

```

3675 \def\eql@keycat{equations,equationsbox,setup}
3676 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
3677 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@if{#3}{#2}{#1}%
3678 {\let\eql@strut@field\eql@strut}{\let\eql@strut@field\relax}}
3679 \eql@define@key\eql@keycat{strutttag}[true]{\eql@decide@if{#3}{#2}{#1}%
3680 {\let\eql@strut@tag\eql@strut}{\let\eql@strut@tag\relax}}

```

TODO: describe **TODO:** maybe also add pre and post variants? for general setup?

```

3681 \eql@define@key{equations}{displaybreak}[4]{\eql@displaybreak@pre{#1}}
3682 \def\eql@keycat{equations,setup}
3683 \eql@define@key\eql@keycat{allowbreaks,allowdisplaybreaks}[4]{%
3684 \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax}
3685 \eql@define@key\eql@keycat{displayheight}[\ht\eql@strutbox@]{%
3686 \def\eql@display@height{#1}}
3687 \eql@define@key\eql@keycat{displaydepth}[\dp\eql@strutbox@]{%

```

```

3688 \def\eqldisplay@depth{#1}
3689 \eql@define@key\eql@keycat{displayheight*}[]{%
3690 \let\eqldisplay@height\undefined}
3691 \eql@define@key\eql@keycat{displaydepth*}[]{%
3692 \let\eqldisplay@depth\undefined}

```

TODO: describe **TODO:** short should just apply to above?! or as far as short would apply...

```

3693 \eql@define@key{equations}{noskip}[]{%
3694 \eql@decide@abovebelow{#3}{#2}{#1}%
3695 {\def\eql@skip@force@above{5}}%
3696 {\def\eql@skip@force@below{5}}}
3697 \eql@define@key{equations}{short}[above]{%
3698 \eql@decide@abovebelow{#3}{#2}{#1}%
3699 {\def\eql@skip@force@above{1}}%
3700 {\def\eql@skip@force@below{1}}}
3701 \eql@define@key{equations}{long}[]{%
3702 \eql@decide@abovebelow{#3}{#2}{#1}%
3703 {\def\eql@skip@force@above{0}}%
3704 {\def\eql@skip@force@below{0}}}
3705 \eql@define@key{equations}{medskip}[]{%
3706 \eql@decide@abovebelow{#3}{#2}{#1}%
3707 {\def\eql@skip@force@above{6}}%
3708 {\def\eql@skip@force@below{6}}}
3709 \eql@define@key{equations}{par}[par]{%
3710 \eql@decide@select{#3}{#2}{#1}{%
3711 {{default,}{\let\eql@skip@force@leave\undefined}},%
3712 {{cont,hmode}{\let\eql@skip@force@leave\z@}},%
3713 {{par,vmode}{\let\eql@skip@force@leave@ne
3714 \ifdefined\eql@skip@force@below\else
3715 \def\eql@skip@force@below{3}%
3716 \fi}},%
3717 {{top}{\let\eql@skip@force@leave\tw@
3718 \ifdefined\eql@skip@force@below\else
3719 \def\eql@skip@force@below{4}
3720 \fi}}}}

```

TODO: describe

```

3721 \eql@define@key{equations}{skip}{%
3722 \def\eql@skip@force@above{7}%
3723 \def\eql@skip@custom@above{#1}%
3724 \let\eql@skip@force@below\eql@skip@force@above
3725 \let\eql@skip@custom@below\eql@skip@custom@above}
3726 \eql@define@key{equations}{aboveskip}{%
3727 \def\eql@skip@force@above{7}%
3728 \def\eql@skip@custom@above{#1}}
3729 \eql@define@key{equations}{belowskip}{%
3730 \def\eql@skip@force@below{7}%
3731 \def\eql@skip@custom@below{#1}}
3732 \eql@define@key{equations}{abovespace}{%
3733 \advance\eql@abovespace@glueexpr#1\relax}
3734 \eql@define@key{equations}{belowspace}{%
3735 \advance\eql@belowspace@glueexpr#1\relax}

```

TODO: describe

```

3736 \eql@define@key{intertext}{skip}{%
3737 \def\eql@skip@force@above{7}%

```

```

3738 \def\eql@skip@custom@above{#1}%
3739 \let\eql@skip@force@below\eql@skip@force@above
3740 \let\eql@skip@custom@below\eql@skip@custom@above}
3741 \eql@define@key{intertext}{aboveskip}{%
3742 \def\eql@skip@force@below{7}%
3743 \def\eql@skip@custom@below{#1}}
3744 \eql@define@key{intertext}{belowskip}{%
3745 \def\eql@skip@force@above{7}%
3746 \def\eql@skip@custom@above{#1}}
3747 \eql@define@key{intertext}{noskip}[]{%
3748 \eql@decide@abovebelow{#3}{#2}{#1}%
3749 {\def\eql@skip@force@below{5}}%
3750 {\def\eql@skip@force@above{5}}}
3751 \eql@define@key{intertext}{short}[]{%
3752 \eql@decide@abovebelow{#3}{#2}{#1}%
3753 {\def\eql@skip@force@below{1}}%
3754 {\def\eql@skip@force@above{1}}}
3755 \eql@define@key{intertext}{long}[]{%
3756 \eql@decide@abovebelow{#3}{#2}{#1}%
3757 {\def\eql@skip@force@below{0}}%
3758 {\def\eql@skip@force@above{0}}}
3759 \eql@define@key{intertext}{medskip}[]{%
3760 \eql@decide@abovebelow{#3}{#2}{#1}%
3761 {\def\eql@skip@force@below{6}}%
3762 {\def\eql@skip@force@above{6}}}

```

TODO: describe

```

3763 \eql@define@key{setup}{skip, longskip}{%
3764 \abovedisplayskip\glueexpr#1\relax
3765 \belowdisplayskip\abovedisplayskip
3766 \def\eql@skip@long@above{#1}%
3767 \let\eql@skip@long@below\eql@skip@long@above}
3768 \eql@define@key{setup}{aboveskip, abovelongskip}{%
3769 \abovedisplayskip\glueexpr#1\relax
3770 \def\eql@skip@long@above{#1}}
3771 \eql@define@key{setup}{belowskip, belowlongskip}{%
3772 \belowdisplayskip\glueexpr#1\relax
3773 \def\eql@skip@long@below{#1}}
3774 \eql@define@key{setup}{aboveshortskip}{%
3775 \abovedisplayshortskip\glueexpr#1\relax
3776 \def\eql@skip@short@above{#1}}
3777 \eql@define@key{setup}{belowshortskip}{%
3778 \belowdisplayshortskip\glueexpr#1\relax
3779 \def\eql@skip@short@below{#1}}
3780 \eql@define@key{setup}{tagskip}{%
3781 \def\eql@skip@tag@above{#1}%
3782 \let\eql@skip@tag@below\eql@skip@tag@above}
3783 \eql@define@key{setup}{abovetagskip}{%
3784 \def\eql@skip@tag@above{#1}}
3785 \eql@define@key{setup}{belowtagskip}{%
3786 \def\eql@skip@tag@below{#1}}
3787 \eql@define@key{setup}{medskip}{%
3788 \def\eql@skip@med@above{#1}%
3789 \let\eql@skip@med@below\eql@skip@med@above}
3790 \eql@define@key{setup}{abovemedskip}{%
3791 \def\eql@skip@med@above{#1}}
3792 \eql@define@key{setup}{belowmedskip}{%
3793 \def\eql@skip@med@below{#1}}

```



```

3794 \eql@define@key{setup}{medtagskip}{%
3795   \def\eql@skip@medtag@above{#1}%
3796   \let\eql@skip@medtag@below\eql@skip@medtag@above}
3797 \eql@define@key{setup}{abovemedtagskip}{%
3798   \def\eql@skip@medtag@above{#1}}
3799 \eql@define@key{setup}{belowmedtagskip}{%
3800   \def\eql@skip@medtag@below{#1}}
3801 \eql@define@key{setup}{abovetopskip}{%
3802   \def\eql@skip@top@above{#1}}
3803 \eql@define@key{setup}{belowtopskip}{%
3804   \def\eql@skip@top@below{#1}}
3805 \eql@define@key{setup}{aboveparskip}{%
3806   \def\eql@skip@par@above{#1}}
3807 \eql@define@key{setup}{belowparskip}{%
3808   \def\eql@skip@par@below{#1}}
3809 \eql@define@key{setup}{abovepartagskip}{%
3810   \def\eql@skip@partag@above{#1}}
3811 \eql@define@key{setup}{belowpartagskip}{%
3812   \def\eql@skip@partag@below{#1}}
3813 \eql@define@key{setup}{abovecontskip}{%
3814   \def\eql@skip@cont@above{#1}}
3815 \eql@define@key{setup}{abovecontskip*}[]{%
3816   \def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}
3817 \eql@define@key{setup}{belowcontskip}{%
3818   \def\eql@skip@cont@below{#1}}
3819 \eql@define@key{setup}{shortmode}{%
3820   \eql@decide@select{#3}{#2}{#1}{%
3821     {off,never,no}{\def\eql@skip@mode@short{0}}},%
3822     {{above,neverbelow,notbelow,belowoff}{\def\eql@skip@mode@short{1}}},%
3823     {{belowone,belowsingle}{\def\eql@skip@mode@short{2}}},%
3824     {{belowall,always,on}{\def\eql@skip@mode@short{3}}}}}}
3825 \eql@define@key{setup}{abovecontmode}{%
3826   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}
3827 \eql@define@key{setup}{belowcontmode}{%
3828   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
3829 \eql@define@key{setup}{aboveparmode}{%
3830   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
3831 \eql@define@key{setup}{belowparmode}{%
3832   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
3833 \eql@define@key{setup}{abovetopmode}{%
3834   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
3835 \eql@define@key{setup}{belowtopmode}{%
3836   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

Labels and Tag Declaration. **TODO:** describe

```

3837 \def\eql@keycat{equations,subequations}
3838 \eql@define@key\eql@keycat{label}{\eql@blocklabel@set{#1}}
3839 \eql@define@key\eql@keycat{tag}{\eql@blocktag@set{#1}}
3840 \eql@define@key\eql@keycat{tag*}{\eql@blocktag@setstar{#1}}

```

Tag Spacing. **TODO:** describe

```

3841 \def\eql@keycat{equations,setup}
3842 \eql@define@key\eql@keycat{tagmargin}[]{\def\eql@tagmargin@val{#1}}
3843 \ifx\eql@tagmargin@val\empty\let\eql@tagmargin@val\undefined\fi}
3844 \eql@define@key\eql@keycat{tagmargin*}{%
3845   \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}}

```

```

3846 \eql@define@key\eql@keycat{tagmarginratio}{%
3847   \eql@tagmargin@ratio@\dimexpr#1pt\relax}
3848 \eql@define@key\eql@keycat{tagmarginthreshold}{%
3849   \def\eql@tagmargin@threshold{#1}}
3850 \eql@define@key\eql@keycat{mintagsep}{\def\eql@tagsepmin@val{#1}}
3851 \eql@define@key\eql@keycat{mintagwidth}{%
3852   \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}}
3853 \eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin@{#1}}

```

Tag Layout. **TODO:** describe

```

3854 \eql@define@key{setup}{tagbox,taglayout}{\eql@tag@setbox{#1}}
3855 \eql@define@key{setup}{tagbox,taglayout*}{\eql@tag@setbox@{#1}}
3856 \eql@define@key{setup}{tagform}{\eql@tag@setform#1}
3857 \eql@define@key{setup}{tagform*}{\eql@tag@setform@{#1}}
3858 \eql@define@key{setup}{subeqtemplate}{%
3859   \def\eql@subequations@template###1###2{#1}%
3860   \expandafter\def\expandafter\eql@subequations@template\expandafter{
3861     \eql@subequations@template\theparentequation{equation}}%
3862 }
3863 %\def\eql@subequations@template{#1}}
3864 \eql@define@key{setup}{autolabel}[true]{%
3865   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@autolabel}
3866 \eql@define@key{setup}{autotag}[true]{%
3867   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@autotag}

```

Equation Numbering. **TODO:** describe

```

3868 \def\eql@keycat{equations,setup}
3869 \eql@define@key\eql@keycat{numberline,numline,n}[all]{%
3870   \eql@numbering@set{##1}}
3871 \eql@define@key\eql@keycat{nonumber,nn,*}[][%
3872   \let\eql@numbering@active\eql@false}
3873 \eql@define@key\eql@keycat{donumber,dn,!}[][%
3874   \let\eql@numbering@active\eql@true}
3875 \eql@define@key\eql@keycat{number,num}[true]{%
3876   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@active}
3877 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
3878 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
3879 \eql@define@key\eql@keycat{tags,eqno}{%
3880   \eql@decide@select{#3}{#2}{#1}{%
3881     {right,r}{\let\eql@tagsleft\eql@false}},%
3882     {left,l}{\let\eql@tagsleft\eql@true}}}}

```

Horizontal Layout. **TODO:** describe

```

3883 \def\eql@keycat{equations,setup}
3884 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
3885   {{center,c}{\let\eql@layoutleft\eql@false}},%
3886   {{left,l}{\let\eql@layoutleft\eql@true}}}}
3887 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
3888 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
3889 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin@val{#1}}
3890 \eql@define@key\eql@keycat{leftmargin*}{%
3891   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin@val{\the\dimen@}}
3892 \eql@define@key\eql@keycat{minleftmargin}{%
3893   \eql@layoutleftmarginmin@\glueexpr#1\relax}

```

```

3894 \eql@define@key\eql@keycat{maxleftmargin}{%
3895   \eql@layoutleftmarginmax@glueexpr#1\relax}
3896 \eql@define@key\eql@keycat{maxleftmargin*}[]{%
3897   \eql@layoutleftmarginmax@.5\maxdimen}

```

Horizontal Spacing and Columns. **TODO:** describe

```

3898 \def\eql@keycat{equations,setup}
3899 \eql@define@key\eql@keycat{marginbadness}{\eql@marginbadness@#1\relax}
3900 \eql@define@key\eql@keycat{maxbadness}{\eql@maxbadness@#1\relax}
3901 \eql@define@key\eql@keycat{mincolsep}{\def\eql@colsepmin@val{#1}}
3902 \eql@define@key\eql@keycat{maxcolsep}{\def\eql@colsepmax@val{#1}}
3903 \eql@define@key\eql@keycat{maxcolsep*}[]{\def\eql@colsepmax@val{.5\maxdimen}}
3904 \eql@define@key\eql@keycat{margins}[true]{%
3905   \eql@decide@bool{#3}{#2}{#1}\eql@columns@margins}
3906 \def\eql@keycat{equationsbox,setup}
3907 \eql@define@key\eql@keycat{margin}{%
3908   \def\eql@box@marginleft{#1}\def\eql@box@marginright{#1}}
3909 \eql@define@key\eql@keycat{marginleft}{\def\eql@box@marginleft{#1}}
3910 \eql@define@key\eql@keycat{marginright}{\def\eql@box@marginright{#1}}

```

Horizontal Shape. **TODO:** describe

```

3911 \def\eql@keycat{equations,equationsbox,setup}
3912 \eql@define@key\eql@keycat{shape}[default]{\eql@shape@set{#1}}
3913 \eql@define@key\eql@keycat{padding,pad}[\eql@indent@val]{%
3914   \let\eql@paddingmax\eql@false
3915   \def\eql@paddingleft@val{#1}\def\eql@paddingright@val{#1}}
3916 \eql@define@key\eql@keycat{padleft}[\eql@indent@val]{%
3917   \let\eql@paddingmax\eql@false\def\eql@paddingleft@val{#1}}
3918 \eql@define@key\eql@keycat{padright}[\eql@indent@val]{%
3919   \let\eql@paddingmax\eql@false\def\eql@paddingright@val{#1}}
3920 \eql@define@key\eql@keycat{padmax}[true]{%
3921   \eql@decide@bool{#3}{#2}{#1}\eql@paddingmax}
3922 \eql@define@key\eql@keycat{indent}[2em]{%
3923   \def\eql@indent@val{#1}}
3924 \eql@define@key\eql@keycat{indent*}[2em]{%
3925   \def\eql@indent@val{#1}\def\eql@paddingleft@val{#1}}

```

Math Classes at Alignment. **TODO:** describe

```

3926 \def\eql@keycat{equations,equationsbox,setup}
3927 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
3928 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
3929 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
3930 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
3931 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
3932 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}{%
3933   {ampeq,amprel,eqafter,beforerel}\eql@class@ampeq},%
3934   {eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}

```

Punctuation. **TODO:** describe

```

3935 \let\eql@punct@main\relax
3936 \def\eql@keycat{equations,equationsbox,setup}
3937 \eql@define@key\eql@keycat{punctsep}[\,]{\def\eql@punct@sep{#1}}
3938 \eql@define@key\eql@keycat{punct}[.]{\def\eql@punct@main{#1}}

```

```

3939 \eql@define@key\eql@keycat{punctline}[,]{\def\eql@punct@line{#1}}
3940 \eql@define@key\eql@keycat{punctcol}[,]{\def\eql@punct@col{#1}}
3941 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
3942 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
3943 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}

```

Global Switches. **TODO:** describe

```

3944 \let\eql@multi@linesfallback\eql@true
3945 \let\eql@single@crerror\eql@true
3946 \let\eql@ampproof@active\eql@false
3947 \eql@define@key{setup}{linesfallback}[true]{%
3948   \eql@decide@bool{#3}{#2}{#1}\eql@multi@linesfallback}
3949 \eql@define@key{setup}{ampproof}[true]{%
3950   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
3951 \eql@define@key{setup}{crerror}[true]{%
3952   \eql@decide@bool{#3}{#2}{#1}\eql@single@crerror}
3953 \eql@define@key{equations,setup}{rescan}[true]{%
3954   \eql@decide@if{#3}{#2}{#1}%
3955   {\let\eql@scan@body\eql@scan@body@rescan}%
3956   {\let\eql@scan@body\eql@scan@body@dump}}
3957 \eql@define@key{setup}{defaults}{%
3958   \eql@decide@select{#3}{#2}{#1}{%
3959     {classic}{\eql@defaults@classic}},%
3960     {eqnlines}{\eql@defaults@eqnlines}}}}

```

Package Options. **TODO:** describe

```

3961 \let\eql@provide@opt@equation\eql@true
3962 \let\eql@provide@opt@amsmathends\eql@true
3963 \let\eql@provide@opt@amsmath\eql@true
3964 \let\eql@provide@opt@ang\eql@true
3965 \let\eql@provide@opt@eqref\eql@true
3966 \eql@define@key{setup}{equation}[true]{%
3967   \eql@error@packageoption{#2}%
3968   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@equation}
3969 \eql@define@key{setup}{amsmathends}[true]{%
3970   \eql@error@packageoption{#2}%
3971   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
3972 \eql@define@key{setup}{amsmath}[true]{%
3973   \eql@error@packageoption{#2}%
3974   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmath}
3975 \eql@define@key{setup}{ang}[true]{%
3976   \eql@error@packageoption{#2}%
3977   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
3978 \eql@define@key{setup}{eqref}[true]{%
3979   \eql@error@packageoption{#2}%
3980   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

P.4 Parameter Sets

TODO: par@above plus parskip?

```

3981 \def\eql@defaults@classic{%
3982   \eqnlineset{mintagsep={.5\fontdimen6\textfont2\relax}}%
3983   \eqnlineset{maxcolsep*}%
3984   \eqnlineset{spread={\jot}}%

```

```

3985 \eqnlineset{tagmargin}%
3986 \eqnlineset{tagmarginratio=1}%
3987 \eqnlineset{tagmarginthreshold=0.5}%
3988 \eqnlineset{leftmargin={\leftmargini}}%
3989 \eqnlineset{displayheight*}%
3990 \eqnlineset{displaydepth*}%
3991 \eqnlineset{shortmode=belowsingle}%
3992 \eqnlineset{abovecontmode=short}%
3993 \eqnlineset{belowcontmode=short}%
3994 \eqnlineset{aboveparmode=long}%
3995 \eqnlineset{belowparmode=long}%
3996 \eqnlineset{abovetopmode=long}%
3997 \eqnlineset{belowtopmode=long}%
3998 \eqnlineset{abovelongskip={\abovedisplayskip}}%
3999 \eqnlineset{belowlongskip={\belowdisplayskip}}%
4000 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
4001 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
4002 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
4003 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
4004 \eqnlineset{abovecontskip=0pt}%
4005 \eqnlineset{belowcontskip=0pt}%
4006 \eqnlineset{aboveparskip=0pt}%
4007 \eqnlineset{belowparskip=0pt}%
4008 \eqnlineset{abovetopskip=0pt}%
4009 \eqnlineset{belowtopskip=0pt}%
4010 \eqnlineset{abovetagskip=0pt}%
4011 \eqnlineset{belowtagskip=0pt}%
4012 \eqnlineset{abovemedtagskip=0pt}%
4013 \eqnlineset{belowmedtagskip=0pt}%
4014 \eqnlineset{abovepartagskip=0pt}%
4015 \eqnlineset{belowpartagskip=0pt}%
4016 }

```

values based on 10pt vs 12pt

```

4017 \def\eql@defaults@eqnlines{%
4018 \eqnlineset{mintagsep=.5em}%
4019 \eqnlineset{maxcolsep=2em}%
4020 \eqnlineset{spread={0.2\normalbaselineskip}}%
4021 \eqnlineset{tagmargin}%
4022 \eqnlineset{tagmarginratio=.334}%
4023 \eqnlineset{tagmarginthreshold=0.5}%
4024 \eqnlineset{leftmargin={\leftmargini}}%
4025 \eqnlineset{displayheight}%
4026 \eqnlineset{displaydepth}%
4027 \eqnlineset{shortmode=above}%
4028 \eqnlineset{abovecontmode=noskip}%
4029 \eqnlineset{belowcontmode=long}%
4030 \eqnlineset{aboveparmode=long}%
4031 \eqnlineset{belowparmode=long}%
4032 \eqnlineset{abovetopmode=noskip}%
4033 \eqnlineset{belowtopmode=long}%
4034 \eqnlineset{longskip={0.75\normalbaselineskip
4035 plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
4036 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
4037 plus 0.25\normalbaselineskip}}%
4038 \eqnlineset{belowshortskip={0.0\normalbaselineskip
4039 plus 0.25\normalbaselineskip}}%
4040 \eqnlineset{medskip={0.4\normalbaselineskip

```

```

4041   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
4042   \eqnlineset{abovecontskip=0pt}%
4043   \eqnlineset{belowcontskip=0pt}%
4044   \eqnlineset{aboveparskip=0pt}%
4045   \eqnlineset{belowparskip=0pt}%
4046   \eqnlineset{abovetopskip=0pt}%
4047   \eqnlineset{belowtopskip=0pt}%
4048   \eqnlineset{abovetagskip=0pt}%
4049   \eqnlineset{belowtagskip=0pt}%
4050   \eqnlineset{abovemedtagskip=0pt}%
4051   \eqnlineset{belowmedtagskip=0pt}%
4052   \eqnlineset{abovepartagskip=0pt}%
4053   \eqnlineset{belowpartagskip=0pt}%
4054 }

```

P.5 Component Selection

TODO: describe

```

4055 \newenvironment{eql@gathered}
4056   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
4057 \newenvironment{eql@multlined}
4058   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
4059 \newenvironment{eql@aligned}
4060   {\eqnaddopt{align}\equationsbox}{\endequationsbox}

```

TODO: describe

```

4061 \newenvironment{eql@equation}
4062   {\eqnaddopt{equation}\equations}{\endequations}
4063 \newenvironment{eql@gather}
4064   {\eqnaddopt{lines}\equations}{\endequations}
4065 \newenvironment{eql@multline}
4066   {\eqnaddopt{lines,padmax,shape=steps,numberline=out}\equations}
4067   {\endequations}
4068 \newenvironment{eql@align}
4069   {\eqnaddopt{align}\equations}{\endequations}
4070 \newenvironment{eql@flalign}
4071   {\eqnaddopt{align,margins=false,maxcolsep*}\equations}{\endequations}
4072 \newenvironment{eql@equation*}
4073   {\eqnaddopt{nonumber}\eql@equation}{\endequations}
4074 \newenvironment{eql@gather*}
4075   {\eqnaddopt{nonumber}\eql@gather}{\endequations}
4076 \newenvironment{eql@multline*}
4077   {\eqnaddopt{nonumber}\eql@multline}{\endequations}
4078 \newenvironment{eql@align*}
4079   {\eqnaddopt{nonumber}\eql@align}{\endequations}
4080 \newenvironment{eql@flalign*}
4081   {\eqnaddopt{nonumber}\eql@flalign}{\endequations}

```

TODO: describe

```

4082 \def\eql@provide@movecmd#1#2{%
4083   \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
4084 }
4085 \def\eql@provide@undefinecmd#1{%
4086   \expandafter\let\csname #1\endcsname\@undefined
4087 }
4088 \def\eql@provide@moveenv#1#2{%

```

```

4089 \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
4090 \expandafter\let\csname end#1\expandafter\endcsname\csname end#2\endcsname
4091 }
4092 \def\eql@provide@undefineenv#1{%
4093 \expandafter\let\csname #1\endcsname\@undefined
4094 \expandafter\let\csname end#1\endcsname\@undefined
4095 }

```

TODO: describe

```

4096 \def\eql@provide@onlyonce#1#2{%
4097 \def\eql@tmp{#2}%
4098 \def\@tempa{#1}%
4099 \ifx\eql@tmp\@tempa
4100 \let\eql@tmp\@undefined
4101 \fi
4102 \ifx\eql@tmp\@empty
4103 \let\eql@tmp\@undefined
4104 \fi
4105 \def\@tempa{*}%
4106 \ifx\eql@tmp\@tempa
4107 \def\eql@tmp{#1}%
4108 \fi
4109 \ifdefined\eql@tmp\else
4110 \ifcsname eql@provided@#1\endcsname
4111 \def\eql@tmp{#1}%
4112 \else
4113 \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4114 \fi
4115 \fi
4116 }

```

TODO: describe

```

4117 \def\eql@provide@cmdonlyonce#1#2{%
4118 \def\eql@tmp{#2}%
4119 \edef\@tempb{\expandafter\noexpand\csname#1\endcsname}%
4120 \ifx\eql@tmp\@tempb
4121 \let\eql@tmp\@undefined
4122 \fi
4123 \ifx\eql@tmp\@empty
4124 \let\eql@tmp\@undefined
4125 \fi
4126 \def\@tempa{*}%
4127 \ifx\eql@tmp\@tempa
4128 \let\eql@tmp\@tempb
4129 \fi
4130 \ifdefined\eql@tmp
4131 \edef\eql@tmp{\expandafter\expandafter\expandafter\@gobble
4132 \expandafter\string\eql@tmp}%
4133 \else
4134 \ifcsname eql@provided@#1\endcsname
4135 \let\eql@tmp\@tempb
4136 \else
4137 \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4138 \fi
4139 \fi
4140 }

```

TODO: describe

```

4141 \def\eql@provide@cmd#1#2{%
4142   \eql@provide@cmdonlyonce{#1}{#2}%
4143   \ifdefined\eql@tmp
4144     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{\eql@#1}%
4145   \else
4146     \eql@amsmath@after{%
4147       \eql@provide@movecmd{ams#1}{#1}%
4148       \eql@provide@movecmd{#1}{\eql@#1}%
4149     }%
4150     \AddToHook{package/mathtools/after}{%
4151       \eql@provide@movecmd{#1}{\eql@#1}%
4152     }%
4153     \eql@provide@movecmd{#1}{\eql@#1}%
4154     \eql@amsmath@before{\eql@provide@undefinecmd{#1}}%
4155   \fi
4156 }

```

TODO: describe

```

4157 \def\eql@amsmath@endfix#1#2{%
4158   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
4159   \long\edef\@tempb{\expandafter\noexpand\csname eql@amsmath@end#2\endcsname}%
4160   \expandafter\ifx\csname end#1\endcsname\@tempa
4161     \expandafter\let\csname end#1\endcsname\@tempb
4162   \fi
4163 }

```

TODO: describe

```

4164 \def\eql@amsmath@fixends{%
4165   \eql@amsmath@after{%
4166     \let\eql@amsmath@endmultiline\endmultiline
4167     \eql@amsmath@endfix{multiline*}{multiline}%
4168     \let\eql@amsmath@endgather\endgather
4169     \eql@amsmath@endfix{gather*}{gather}%
4170     \let\eql@amsmath@endalign\endalign
4171     \eql@amsmath@endfix{align*}{align}%
4172     \eql@amsmath@endfix{flalign}{align}%
4173     \eql@amsmath@endfix{flalign*}{align}%
4174     \eql@amsmath@endfix{alignat}{align}%
4175     \eql@amsmath@endfix{alignat*}{align}%
4176     \eql@amsmath@endfix{xalignat}{align}%
4177     \eql@amsmath@endfix{xalignat*}{align}%
4178     \eql@amsmath@endfix{xxalignat}{align}%
4179     \let\eql@amsmath@endaligned\endaligned
4180     \eql@amsmath@endfix{gathered}{aligned}%
4181     \eql@amsmath@endfix{alignedat}{aligned}%
4182   }
4183 }

```

TODO: describe

```

4184 \def\eql@provide@env#1#2{%
4185   \eql@provide@onlyonce{#1}{#2}%
4186   \ifdefined\eql@tmp
4187     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@#1}%
4188   \else
4189     \eql@amsmath@after{%
4190       \eql@provide@moveenv{ams#1}{#1}%
4191       \eql@provide@moveenv{ams#1*}{#1*}%
4192       \eql@provide@moveenv{#1}{\eql@#1}%

```



```

4193     \eql@provide@moveenv{#1*}{eql@#1*}%
4194 }%
4195 \AddToHook{package/mathtools/after}{%
4196     \eql@provide@moveenv{#1}{eql@#1}%
4197     \eql@provide@moveenv{#1*}{eql@#1*}%
4198 }%
4199 \eql@provide@moveenv{#1}{eql@#1}%
4200 \eql@provide@moveenv{#1*}{eql@#1*}%
4201 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
4202 \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
4203 \fi
4204 }

```

TODO: describe

```

4205 \def\eql@provide@env@equation#1{%
4206     \eql@provide@onlyonce{equation}{#1}%
4207     \ifdefined\eql@tmp
4208         \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@equation}%
4209     \else
4210         \eql@amsmath@after{%
4211             \eql@provide@moveenv{amsequation}{equation}%
4212             \eql@provide@moveenv{amsequation*}{equation*}%
4213             \eql@provide@moveenv{equation}{eql@equation}%
4214             \eql@provide@moveenv{equation*}{eql@equation*}%
4215         }%
4216         \AddToHook{package/hyperref/after}{%
4217             \@ifpackageloaded{amsmath}{}{-%
4218                 \let\latexequation\H@equation
4219                 \let\endlatexequation\H@endequation
4220                 \eql@provide@moveenv{hyperrefequation}{equation}%
4221                 \eql@provide@moveenv{equation}{eql@equation}%
4222             }%
4223         }%
4224         \@ifpackageloaded{amsmath}{}{\@ifpackageloaded{hyperref}{}{-%
4225             \eql@provide@moveenv{latexequation}{equation}%
4226         }}%
4227         \eql@provide@moveenv{equation}{eql@equation}%
4228         \eql@provide@moveenv{equation*}{eql@equation*}%
4229         \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
4230     \ifdefined\eql@tagging@on
4231         \AddToHook{begindocument/end}{%
4232             \eql@provide@moveenv{equation}{eql@equation}%
4233             \eql@provide@moveenv{equation*}{eql@equation*}%
4234         }%
4235     \fi
4236 \fi
4237 }

```

TODO: describe

```

4238 \def\eql@provide@env@multlined#1{%
4239     \eql@provide@onlyonce{multlined}{#1}%
4240     \ifdefined\eql@tmp
4241         \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
4242     \else
4243         \AddToHook{package/mathtools/after}{%
4244             \eql@provide@moveenv{amsmultlined}{multlined}%
4245             \eql@provide@moveenv{multlined}{eql@multlined}%
4246         }%

```

```

4247 \eql@provide@moveenv{multlined}{eql@multlined}%
4248 \@ifpackageloaded{mathtools}{-}{\AddToHook{package/mathtools/before}{%
4249 \eql@provide@undefineenv{multlined}}}%
4250 \fi
4251 }

```

TODO: describe

```

4252 \def\eql@provide@env@subequations#1{%
4253 \eql@provide@onlyonce{subequations}{#1}%
4254 \ifdefined\eql@tmp
4255 \expandafter\eql@provide@moveenv
4256 \expandafter{\eql@tmp}{eql@subequations}%
4257 \else
4258 \eql@amsmath@after{%
4259 \eql@provide@moveenv{amssubequations}{subequations}%
4260 \eql@provide@moveenv{subequations}{eql@subequations}%
4261 }%
4262 \AddToHook{package/hyperref/after}{%
4263 \AddToHook{cmd/subequations/before}[hyperref]{}%
4264 \AddToHook{cmd/subequations/after}[hyperref]{}%
4265 \RemoveFromHook{cmd/subequations/before}[hyperref]%
4266 \RemoveFromHook{cmd/subequations/after}[hyperref]%
4267 \AddToHook{cmd/amssubequations/before}%
4268 {%
4269 \stepcounter{equation}%
4270 \protected@edef\theHparentequation{\theHequation}%
4271 \addtocounter{equation}{-1}%
4272 }%
4273 \AddToHook{cmd/amssubequations/after}%
4274 {%
4275 \def\theHequation{\theHparentequation\alph{equation}}%
4276 \ignorespaces
4277 }%
4278 \AddToHook{begindocument/end}{%
4279 \eql@provide@moveenv{subequations}{eql@subequations}%
4280 }%
4281 }%
4282 \eql@provide@moveenv{subequations}{eql@subequations}%
4283 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
4284 \fi
4285 }

```

TODO: describe

```

4286 \def\eql@provide@sqr{%
4287 \let\[ \eql@equations@sqr@open
4288 \let\] \eql@equations@sqr@close
4289 \eql@amsmath@after{%
4290 \let\[ \eql@equations@sqr@open
4291 \let\] \eql@equations@sqr@close
4292 }%
4293 \ifdefined\eql@tagging@on
4294 \AddToHook{begindocument/end}{%
4295 \let\[ \eql@equations@sqr@open
4296 \let\] \eql@equations@sqr@close
4297 }%
4298 \fi
4299 }

```

TODO: describe

```
4300 \def\eql@provide@ang{%
4301   \let\<\eql@equations@ang@open
4302   \let\>\eql@equations@ang@close
4303 }
```

TODO: describe

```
4304 \eql@define@key{provide}{equation} [] {\eql@provide@env@equation{#1}}
4305 \eql@define@key{provide}{gather} [] {\eql@provide@env@gather{#1}}
4306 \eql@define@key{provide}{multline} [] {\eql@provide@env@multline{#1}}
4307 \eql@define@key{provide}{align} [] {\eql@provide@env@align{#1}}
4308 \eql@define@key{provide}{flalign} [] {\eql@provide@env@flalign{#1}}
4309 \eql@define@key{provide}{aligned} [] {\eql@provide@env@aligned{#1}}
4310 \eql@define@key{provide}{gathered} [] {\eql@provide@env@gathered{#1}}
4311 \eql@define@key{provide}{multlined} [] {\eql@provide@env@multlined{#1}}
4312 \eql@define@key{provide}{subequations} [] {\eql@provide@env@subequations{#1}}
4313 \eql@define@key{provide}{sqr} [] {\eql@provide@sqr}
4314 \eql@define@key{provide}{ang} [] {\eql@provide@ang}
4315 \eql@define@key{provide}{eqref} [] {\eql@provide@cmd{eqref}{#1}}
4316 \eql@define@key{provide}{tagform} [] {%
4317   \def\tagform@##1{\maketag@@@{\eql@tag@form{#1}}}}
4318 \eql@define@key{provide}{maketag} [] {%
4319   \def\maketag@@@##1{\eql@tag@box{##1}}}
```

TODO: describe

```
4320 \newcommand{\eqnlinesprovide}[1]{%
4321 (dev)\eql@dev@start\eqnlinesprovide
4322   \eql@setkeys{provide}{#1}}
```

P.6 Global and Package Options

Handle global and package options:

`\eqnlineset` The macro `\eqnlineset` processes global configuration options including the package options:

```
4323 \newcommand{\eqnlineset}[1]{%
4324 (dev)\eql@dev@start\eqnlineset
4325   \eql@setkeys{setup}{#1}}
```

Disable error message for exclusive package options:

```
4326 \let\eql@error@packageoption@gobble
```

Declare math layout options `leqno` and `fleqn` for common L^AT_EX classes:

```
4327 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
4328 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
4329 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
4330 \eql@defaults@eqnlines
4331 \eql@mode@columns
4332 \eql@mode@aligned
```

Process package options:

```
4333 \ProcessOptions
```

`\error@packageoption` Enable error message for exclusive package options:

```
4334 \def\eql@error@packageoption#1{%
4335   \eql@error{may only use ‘#1’ as a package option}%
4336 }
```

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:

```
4337 \ifdefined\tagsleft@true\else
4338   \expandafter\newif\csname iftagsleft@\endcsname
4339 \fi
4340 \ifdefined\@fleqntrue\else
4341   \expandafter\newif\csname if@fleqn\endcsname
4342 \fi
```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`:

```
4343 \ifdefined\eql@provide@opt@amsmath
4344   \let\eql@provide@opt@equation\eql@true
4345   \eql@amsmath@after{%
4346     \iftagsleft@
4347       \eqnlineset{tagsleft}
4348     \else
4349       \eqnlineset{tagsright}
4350     \fi
4351     \if@fleqn
4352       \eqnlineset{left}
4353     \else
4354       \eqnlineset{center}
4355     \fi
4356   }
4357 \fi
```

Provide native `LATEX` environment `equation` and symbolic shortcut `\[...\]` if desired:

```
4358 \ifdefined\eql@provide@opt@equation\eqnlinesprovide{equation,sqr}\fi
```

Make the ending statements for `amsmath` environments independent if desired, so that they may be overwritten individually:

```
4359 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

Provide `amsmath` equation environments if desired:

```
4360 \ifdefined\eql@provide@opt@amsmath
4361   \eqnlinesprovide{%
4362     multiline,gather,align,flalign,%
4363     multlined,gathered,aligned,%
4364     subequations}
4365 \fi
```

Provide symbolic shortcut `\<...\>` if desired:

```
4366 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
4367 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```