

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`

2025/04/09, v0.7.1

<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	23
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	28
C.1	Parameters	28
C.2	Registers	31
C.3	Hooks	33
D	Punctuation	33
E	Math Classes at Alignment	35
F	Equation Numbering	35
F.1	Tag Formatting	35
F.2	Showkeys Integration	36
F.3	Labels	37
F.4	Tags	38
F.5	Anchors	39
F.6	Tag Composition	39
F.7	Tagbox Methods	41
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	51
H	Display Equations Support	51
H.1	Display Breaks	51
H.2	General Initialisation	52
H.3	halign Support	56
H.4	Stack	58
I	Horizontal Spacing for Lines	59
I.1	Supporting Definitionss	59
I.2	Shape Schemes	60
I.3	Adjustment Methods	63
I.4	Central Alignment Layout	65
I.5	Left Alignment Layout	70
J	Multi-Line Support	73
J.1	Table Registers	73
J.2	Measure Support	74
J.3	Print Support	75
J.4	Line Breaks	76
J.5	Intertext	76
K	Equations Box Environment	78

K.1	Line Breaks	78
K.2	Stacked Mode	79
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	86
N	Multi-Line Columns Mode	87
N.1	Columns Processing	87
N.2	Measure	87
N.3	Print	90
N.4	Adjust	94
O	Interface	100
O.1	Scanning the Equation Body	100
O.2	Options Processing	103
O.3	Single-Line Main	103
O.4	Multi-Line Main	105
O.5	Equations Environment	106
P	Options	108
P.1	Selection Tools	108
P.2	Declaration Code	109
P.3	Options Declarations	110
P.4	Parameter Sets	117
P.5	Component Selection	118
P.6	Global and Package Options	124

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 v0.7.1: 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.*) `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 [...]):`

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

`\dn,!` (*key*)

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

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

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

`\numbernext`

`\numberhere` and `\numbernext`

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

<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 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 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.

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

`\eqnpunct` `punct` (*key*)
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 `punctcol` (*key*) 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: $\lceil \mbox{long mode} \rceil$ vs. \short: $\lceil \mbox{short mode} \rceil$ 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
off	disabled
above	above short equations (package default)
belowone	also below short single-line equations
belowall	also below all short multi-line equations

short (*key*) 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: $\lceil\lceil\short\rceil\mbox{forced short}\rceil$ and short: $\lceil\lceil\long\rceil\mbox{forced long}\rceil$ 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>$\hrule\begin{minipage}{\linewidth}$ $\lceil \mbox{top} \rceil$ some text\par $\lceil \mbox{par} \rceil$ $\lceil \mbox{cont} \rceil$ $\end{minipage}\hrule$</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 and modifiers, but it offers a reduced range of functionality such as (evidently) no numbering (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 \>

```

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

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>	<code>eqnlines</code> layout with fontsize-relative dimensions
<code>rescan</code>	rescan environment body for special commands (e.g. <code>\verb</code>)
<code>linesfallback</code>	single column in align mode reverts to lines mode
<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>backup</code>	bool	backup original <code>amsmath</code> environments as <code>ams...</code>
<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` overwrites most math environments of `amsmath` with its own implementations. It can also preserve them as `ams...` if needed. Alternatively, `eqnlines` may assign individual names to the maths environments and preserve the ones of `amsmath`. The other features provided by `amsmath` can be used.
- The package `mathtools` is a popular extension of the `amsmath` package. This package incorporates some of the features and improvements provided by the `mathtools` package. Compatibility with the `mathtools` package has been tested with v1.31 (2024/10/04), and it is maintained whether `eqnlines` is loaded before or after `mathtools`. Some features like 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

v0.7.1: 2025/04/09

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

v0.7: 2025/04/03

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

v0.6.1: 2025/03/27

- `\eqnpunct` can place punctuation within the current equation 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 `eqlines-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\eqld@dev#1{\PackageInfo{eqlines-dev}{#1}}
2 <dev>\def\eqld@dev@start#1{\eqld@dev{starting \string#1}}
3 <dev>\def\eqld@dev@enter#1{\eqld@dev{entering \string#1}}
4 <dev>\def\eqld@dev@leave#1{\eqld@dev{ leaving \string#1}}
5 <dev>\def\eqld@dev@enterenv{\eqld@dev{entering \@currrent}}
6 <dev>\def\eqld@dev@leaveenv{\eqld@dev{ leaving \@currrent}}
7 <dev>\def\eqld@dev@in#1#2{\eqld@dev{ \space within \string#1 #2}}
```

B.2 Supporting Definitions

`\eqld@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 `\ifdefinedxxx`. In order to make the scheme more tangible, we define the two expected values for boolean variables:

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

B.3 Dollardollar Abstraction

`\dollar@dollar@begin` As of 2025 L^AT_EX defines `\dollar@dollar@begin` and `\dollar@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@dollar@begin
11 \def\eqld@dollar@dollar@begin{\dollar@dollar@begin}
12 \def\eqld@dollar@dollar@end{\dollar@dollar@end}
13 \else
14 \def\eqld@dollar@dollar@begin{$$$}
15 \def\eqld@dollar@dollar@end{$$$}
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.

`\eqld@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 `\eqld@ifnextchar@loose`. We replicate

the `\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\eql@ifnch@tight
23 }
24 \def\eql@ifnch@tight{%
25   \ifx\@let@token\reserved@d
26     \let\reserved@b\reserved@a
27   \fi
28   \reserved@b
29 }

```

`\eql@atxii` Capture ‘@’ as a character (catcode 12) rather than a letter (catcode 11) as `\eql@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\eql@atxii\tmp
35 \endgroup

```

`\eql@ifnextgobble@...` We introduce a collection of look-ahead macros which do or do not skip over spaces. The `\eql@ifstar@...` macros `\eql@ifstar@...` and `\eql@testopt@...` replicate the L^AT_EX counterparts `\@ifstar` and `\@testopt`. The macros `\eql@ifnextgobble@...` work like `\@ifnextchar`, `\eql@teststaropt@...` but also gobble the specific character if found; one might define `\eql@ifstar@...` as `\eql@ifnextgobble@...*`. The macros `\eql@teststaropt@...` tests for combinations of ‘*’ and optional arguments [...]:

```

36 \long\def\eql@ifnextgobble@loose#1#2{\eql@ifnextchar@loose#1{\@firstoftwo{#2}}}
37 \long\def\eql@ifnextgobble@tight#1#2{\eql@ifnextchar@tight#1{\@firstoftwo{#2}}}
38 \long\def\eql@ifstar@loose#1{\eql@ifnextchar@loose*\@firstoftwo{#1}}
39 \long\def\eql@ifstar@tight#1{\eql@ifnextchar@tight*\@firstoftwo{#1}}
40 \long\def\eql@ifat@loose#1#2{\eql@ifnextgobble@loose{#1}{%
41   \eql@ifnextgobble@loose\eql@atxii{#1}{#2}}}
42 \long\def\eql@ifat@tight#1#2{\eql@ifnextgobble@tight{#1}{%
43   \eql@ifnextgobble@tight\eql@atxii{#1}{#2}}}
44 \long\def\eql@testopt@loose#1#2{\eql@ifnextchar@loose[#{1}]{#1[#{2}]}%]}
45 \long\def\eql@testopt@tight#1#2{\eql@ifnextchar@tight[#{1}]{#1[#{2}]}%]}
46 \long\def\eql@teststaropt@loose#1#2#3{%
47   \eql@ifstar@loose{\eql@testopt@loose{#1}{#3}}{\eql@testopt@loose{#2}{#3}}
48 \long\def\eql@teststaropt@tight#1#2#3{%
49   \eql@ifstar@tight{\eql@testopt@tight{#1}{#3}}{\eql@testopt@tight{#2}{#3}}

```

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

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>.


```

50 \def\eq@srbgroup{\iffalse{\fi\ifnum0='}\fi}
51 \def\eq@spegroup{\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}

```

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

```

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

```

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

```

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

```

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

```

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

```

B.5 Error Messages

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

```

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

```

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

`\eq@error@mathmode`

```

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

```

```

\eq@warn@label@unused Warning messages concerning unused and multiply declared labels and tags:
\warn@label@multiple
\eq@warn@tag@unused 85 \def\eq@warn@label@unused{\eq@warning{Unused equation \string\label:
\eq@warn@tag@multiple 86 label '\eq@nextlabel' will be lost}}
87 \def\eq@warn@label@multiple#1{\eq@warning{Multiple equation \string\label's:
88 previous label '#1' will be lost}}
89 \def\eq@warn@tag@unused{\eq@warning{Unused equation \string\tag:
90 tag declaration will be lost}}
91 \def\eq@warn@tag@multiple{\eq@warning{Multiple equation \string\tag's:
92 previous tag declaration will be lost}}

```

B.6 amsmath Integration

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

```

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

```

TODO: temporary fix for development stages

```

98 \@ifpackageloaded{amsmath}{\{
99   \DeclareHookRule{package/amsmath/after}
100   {eqnlines}{after}{latex-lab-testphase-math}}

```

B.7 PDF Tagging Support

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

```

101 \let\eq@tagging@on\eq@false
102 \IfFormatAtLeastTF{2025-06-01}{%
103   \csname tag_if_active:T\endcsname{\let\eq@tagging@on\eq@true}}{\}
104 \ifdefined\eq@tagging@on
105   \def\eq@tagging@mathsave{%
106     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}
107   \def\eq@tagging@mathaddlast{%
108     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
109   \def\eq@tagging@tagbegin{%
110     \UseTaggingSocket{math/display/tag/begin}}
111   \def\eq@tagging@tagend{%
112     \UseTaggingSocket{math/display/tag/end}}
113   \def\eq@tagging@tagsave{%
114     \UseTaggingSocket{math/luamml/mtable/tag/save}}
115   \def\eq@tagging@tagaddbox{%
116     \setbox\z@\copy\eq@tagbox%
117     \UseTaggingSocket{math/luamml/mtable/tag/set}}
118   \def\eq@tagging@tablesaveinner{%

```

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

```

119   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
120 \def\eql@tagging@tableaddinner{%
121   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
122 \def\eql@tagging@tablesavelines{%
123   \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
124 \def\eql@tagging@tablesavealign{%
125   \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
126 \def\eql@tagging@alignleft{%
127   \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
128 \def\eql@tagging@aligncenter{%
129   \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
130 \def\eql@tagging@alignright{%
131   \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:

```

132 \let\eql@single@doscan\eql@true
133 \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:

```

```

134 \ExplSyntaxOn
135 \def\eql@tagging@start{%
136   \bool_if:NF\l__math_collected_bool{%
137     \toks@\expandafter{\eql@tagging@opt}%
138     \edef\eql@tmp{\@currentenv}{[\the\toks@] \the\eql@scan@reg}}%
139     \expandafter\__math_process:nn\eql@tmp
140     \@kernel@math@registered@begin
141     \bool_set_true:N\l__math_collected_bool
142   }%
143 }
144 \def\eql@tagging@end{}
145 \def\eql@tagging@register@env{\math_register_env:n}
146 \ExplSyntaxOff
147 \else
148 \def\eql@tagging@mathsave{}
149 \def\eql@tagging@mathaddlast{}
150 \def\eql@tagging@tagbegin{}
151 \def\eql@tagging@tagend{}
152 \def\eql@tagging@tagsave{}
153 \def\eql@tagging@tagaddbox{}
154 \def\eql@tagging@tablesaveinner{}
155 \def\eql@tagging@tableaddinner{}
156 \def\eql@tagging@tablesavelines{}
157 \def\eql@tagging@tablesavealign{}
158 \def\eql@tagging@alignleft{}
159 \def\eql@tagging@aligncenter{}
160 \def\eql@tagging@alignright{}
161 \def\eql@tagging@start{}
162 \def\eql@tagging@end{}
163 \def\eql@tagging@register@env{\@gobble}
164 \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:

```
165 \let\eql@tagsleft\eql@false
```

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

```
166 \let\eql@layoutleft\eql@false
```

`\eql@leftmargin@` (*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@leftmargin@ (dimen)
\layoutleftmargin@val
\leftmarginmin@ (dimen)
\leftmarginmax@ (dimen)
167 \newdimen\eql@layoutleftmargin@
168 \newdimen\eql@layoutleftmarginmin@
169 \newdimen\eql@layoutleftmarginmax@
170 \def\eql@layoutleftmargin@val{\leftmargini}
171 \eql@layoutleftmarginmin@\z@
172 \eql@layoutleftmarginmax@.5\maxdimen
```

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

```
173 \newdimen\eql@tagmargin@
174 \let\eql@tagmargin@val\@undefined
175 \newdimen\eql@tagmargin@ratio@
176 \eql@tagmargin@ratio@\p@
177 \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:

```
178 \newdimen\eql@indent@
179 \def\eql@indent@val{2em}
```

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

```
180 \newdimen\eql@paddingleft@
181 \newdimen\eql@paddingright@
```

```
182 \def\eql@paddingleft@val{0pt}
183 \def\eql@paddingright@val{0pt}
```

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

```
184 \let\eql@paddingmax\eql@false
```

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

```
185 \def\eql@box@marginleft{\z@skip}
186 \def\eql@box@marginright{\z@skip}
```

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

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

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

```
188 \def\eql@spread@val{\jot}
189 \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.

```
190 \newdimen\eql@tagfuzz@
191 \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:

```
192 \def\eql@display@height{\ht\eql@strutbox@}
193 \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:

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

195 \def\eql@skip@mode@cont@above{2}
196 \def\eql@skip@mode@cont@below{0}

197 \def\eql@skip@mode@par@above{3}
198 \def\eql@skip@mode@par@below{0}

199 \def\eql@skip@mode@top@above{4}
200 \def\eql@skip@mode@top@below{0}

201 \newcount\eql@skip@mode@leave@
202 \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
\mode@above@ (counter) 203 \newcount\eql@skip@mode@above@
\mode@below@ (counter) 204 \newcount\eql@skip@mode@below@
205 \let\eql@skip@force@above@\undefined
206 \let\eql@skip@force@below@\undefined
207 \let\eql@skip@custom@above@\undefined
208 \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@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@above` and `\eql@skip@med@below`:

```
209 \def\eql@skip@long@above{\abovedisplayskip}
210 \def\eql@skip@long@below{\belowdisplayskip}
211 \def\eql@skip@short@above{\abovedisplayshortskip}
212 \def\eql@skip@short@below{\belowdisplayshortskip}
213 \def\eql@skip@cont@above{\eql@skip@short@above}
214 \def\eql@skip@cont@below{\eql@skip@short@below}
215 % \TODO: parabove plus parskip?
216 \def\eql@skip@par@above{\eql@skip@long@above}
217 \def\eql@skip@par@below{\eql@skip@long@below}
218 \def\eql@skip@top@above{\eql@skip@long@above}
219 \def\eql@skip@top@below{\eql@skip@long@below}
220 \def\eql@skip@med@above{\abovedisplayskip/2}
221 \def\eql@skip@med@below{\belowdisplayskip/2}
222 \def\eql@skip@tag@above{\z@skip}
223 \def\eql@skip@tag@below{\z@skip}
224 \def\eql@skip@partag@above{\z@skip}
225 \def\eql@skip@partag@below{\z@skip}
226 \def\eql@skip@medtag@above{\z@skip}
227 \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:

```
228 \newdimen\eql@colsepmin@
229 \def\eql@colsepmin@val{1em}
230 \def\eql@colsepmax@val{.5\maxdimen}
```

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

```
231 \newdimen\eql@tagwidthmin@
232 \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.

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

```

233 \newdimen\eql@tagsepmin@
234 \def\eql@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

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

```

235 \def\eql@equations@sqr@opt{equation,nonumber}
236 \def\eql@equations@ang@opt{align,nonumber}

```

Multi-Line Align Mode.

```

237 \let\eql@columns@margins\eql@true

```

C.2 Registers

TODO: describe

General.

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

```

238 \newbox\eql@fieldbox@
239 \newbox\eql@tagbox@
240 \newdimen\eql@fieldwidth@
241 \newdimen\eql@tagwidth@

```

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

`\eql@tagwidth@max@` (*dimen*)

```

242 \newdimen\eql@totalwidth@
243 \newdimen\eql@tagwidth@max@

```

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

```

244 \newdimen\eql@line@height@
245 \newdimen\eql@line@depth@

```

`\eql@line@width@` (*dimen*)

`\eql@line@avail@` (*dimen*)

`\eql@line@pos@` (*dimen*)

```

246 \newdimen\eql@line@width@
247 \newdimen\eql@line@avail@
248 \newdimen\eql@line@pos@

```

Rows and Columns.

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

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

```

249 \newcount\eql@row@
250 \newcount\eql@totalrows@
251 \newcount\eql@tagrows@

```

`\eql@column@`
`\eql@totalcolumns@` 252 `\newcount\eql@column@`
253 `\newcount\eql@totalcolumns@`

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

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

Vertical Spacing Adjustments.

`\eql@firstavail@` (*dimen*) The unused space on the first line of an alignment is stored in `\eql@display@firstavail@` for comparison against `\predisplaysize` and determining short skip mode of display equations. It is convenient to set it via `\eql@display@firstavail@set` provided that we are on the first line:
256 `\newdimen\eql@display@firstavail@`
257 `\def\eql@display@firstavail@set#1{%`
258 `\ifnum\eql@row@=\@one`
259 `\global\eql@display@firstavail@#1%`
260 `\fi`
261 `}`

`\eql@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:
262 `\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:

```
263 \ifdefined\measuring@true\else
264 \expandafter\newif\csname ifmeasuring@\endcsname
265 \fi
266 \AddToHook{package/hyperref/after}{
267 \ifdefined\Hy@ifnotmeasuring
268 \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
269 \fi
270 }
```

`\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:

```
271 \ifdefined\@displaytrue\else
272 \expandafter\newif\csname if@display\endcsname
273 \everydisplay\expandafter{\the\everydisplay\@displaytrue}
274 \fi
```


C.3 Hooks

`\eql@hook@...` For what it's worth, we define a couple of entry points where one might hook into the equations typesetting framework. The L^AT_EX 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

```
275 \let\eql@hook@blockbefore\@empty
276 \let\eql@hook@blockafter\@empty
277 \let\eql@hook@blockin\@empty
278 \let\eql@hook@blockout\@empty
279 \let\eql@hook@linein\@empty
280 \let\eql@hook@lineout\@empty
281 \let\eql@hook@colin\@empty
282 \let\eql@hook@colout\@empty
283 \let\eql@hook@eqin\@empty
284 \let\eql@hook@eqout\@empty
285 \let\eql@hook@innerleft\@empty
286 \let\eql@hook@innerright\@empty
287 \let\eql@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.

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

```
288 \let\eql@punct@col\@empty
289 \let\eql@punct@line\relax
290 \let\eql@punct@block\relax
```

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

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

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

```
\eqnpunct
292 \def\eqnpunctcol{\eql@ifstar@tight\eql@punct@col@setrelax\eql@punct@col@set}
293 \def\eql@punct@col@set#1{\def\eql@punct@col{#1}\ignorespaces}
294 \def\eql@punct@col@setrelax{\let\eql@punct@col\@empty\ignorespaces}
295 \def\eqnpunctline{\eql@ifstar@tight\eql@punct@line@setrelax\eql@punct@line@set}
296 \def\eql@punct@line@set#1{\def\eql@punct@line{#1}\ignorespaces}
297 \def\eql@punct@line@setrelax{\let\eql@punct@line\relax\ignorespaces}
298 \def\eqnpunctmain{\eql@ifstar@tight\eql@punct@main@setrelax\eql@punct@main@set}
299 \def\eql@punct@main@set#1{\def\eql@punct@main{#1}\ignorespaces}
300 \def\eql@punct@main@setrelax{\let\eql@punct@main\relax\ignorespaces}
301 \def\eqnpunct{\eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
302 \def\eql@punct@next@set#1{%
303   \ifmmode
304     \def\eql@punct@col{#1}%
305     \def\eql@punct@line{#1}%
306     \def\eql@punct@block{#1}%
```

```

307 \else
308   \eqnaddopt{punct={#1}}%
309 \fi
310 \ignorespaces}
311 \def\eql@punct@next@setrelax{%
312 \ifmmode
313   \let\eql@punct@block\relax
314 \else
315   \eqnaddopt{punct*}%
316 \fi
317 \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:

```

318 \def\eql@punct@apply@col{%
319 \ifx\eql@punct@col\@empty\else
320   \eql@punct@sep
321   \eql@punct@col
322   \let\eql@punct@col\@empty
323 \fi
324 }

```

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`

```

325 \def\eql@punct@apply@line{%
326 \ifx\eql@punct@line\relax
327 % \TODO hand down immediately?
328 \else
329   \ifx\eql@punct@line\@empty\else
330     \eql@punct@sep
331     \eql@punct@line
332   \fi
333   \let\eql@punct@line\relax
334   \let\eql@punct@col\@empty
335 \fi
336 }

```

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

```

337 \def\eql@punct@apply@block{%
338 \ifx\eql@punct@block\relax
339 % \TODO hand down immediately?
340 \else
341   \ifx\eql@punct@block\@empty\else
342     \eql@punct@sep
343     \eql@punct@block
344   \fi
345   \let\eql@punct@block\relax
346   \let\eql@punct@line\relax
347   \let\eql@punct@col\@empty
348 \fi
349 }

```

```
350 \let\eqnpunctapply\eql@punct@apply@block
```

E Math Classes at Alignment

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

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

```
351 \def\eql@class@innerright@sel@{%
352   \ifdim\eql@fieldwidth@=\z@
353     \eql@class@innerlead
354   \else
355     \eql@class@innerright
356   \fi
357 }
```

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

`\eql@class@innerright@set`

`\eql@class@innerlead@set`

```
358 \def\eql@class@innerleft@set#1{%
359   \def\eql@class@innerleft{#1}%
360 }
361 \def\eql@class@innerright@set#1{%
362   \def\eql@class@innerright{#1}%
363   \let\eql@class@innerright@sel\eql@class@innerright
364 }
365 \def\eql@class@innerlead@set#1{%
366   \def\eql@class@innerlead{#1}%
367   \let\eql@class@innerright@sel\eql@class@innerright@sel@
368 }
```

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

```
369 \def\eql@class@ampeq{%
370   \eql@class@innerleft@set{}%
371   \eql@class@innerright@set{}}%
372 }
373 \def\eql@class@eqamp{%
374   \eql@class@innerleft@set{\mathrel{}}%
375   \eql@class@innerright@set{\mathrel{}}%
376   \eql@class@innerlead@set{}}%
377 }
378 \eql@class@ampeq
```

F Equation Numbering

TODO: describe

F.1 Tag Formatting

TODO: describe

```

379 \def\eq@tag@setbox@#1{%
380   \def\eq@tag@box##1{#1}%
381 }
382 \def\eq@tag@setbox#1{%
383   \def\eq@tag@box##1{\hbox{\m@th\normalfont#1}}%
384 }

```

TODO: describe

```

385 \def\eq@tag@setform@#1{%
386   \def\eq@tag@form##1{#1}%
387 }
388 \def\eq@tag@setform#1#2#3{%
389   \def\eq@tag@form##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
390 }

```

```

391 \eq@tag@setbox{#1}
392 \eq@tag@setform{#1}
393 \def\eq@tag@boxedform#1{\eq@tag@box{\eq@tag@form{#1}}}

394 \DeclareRobustCommand{\tagform}{\eq@tag@form}
395 \DeclareRobustCommand{\tagbox}{\eq@tag@box}
396 \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:

```

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

```

Raise Tags.

`\settag@amount@` (*dimen*)

```

398 \newdimen\eq@raisetag@amount@

```

`\raisetag`

```

399 \def\eq@raisetag@default{%
400   \eq@warning{\string\raisetag\space not allowed here}
401   \@gobble
402 }

```

TODO: describe

```

403 \eq@amsmath@let\raisetag\eq@raisetag@default

```

TODO: maybe introduce a star form to enforce raise?

```

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

405 \let\eq@raisetag@gobble\@gobble

```

F.2 Showkeys Integration

TODO: describe

```

406 \let\eq@SK@loaded\eq@false
407 \let\eq@SK@label\@gobble
408 \let\eq@SK@clearlabel\@empty

```

```

409 \let\eqL@SK@setlabel\@gobble
410 \let\eqL@SK@printlabel@right\@empty
411 \let\eqL@SK@printlabel@left\@empty
412 \let\eqL@SK@printlabel@line\@empty
413 \def\eqL@label@clean{\eqL@label@org}
414 \AddToHook{package/showkeys/after}{
415   \let\eqL@SK@loaded\eqL@true
416   \def\eqL@SK@label#1{\SK@\SK@@label#1}
417   \def\eqL@SK@clearlabel{\let\eqL@SK@lab\relax}
418   \eqL@SK@clearlabel
419   \def\eqL@SK@@label#1>#2\SK@{%
420     \def\eqL@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
421   }
422   \def\eqL@SK@setlabel#1{\SK@\eqL@SK@@label#1}
423   \def\eqL@SK@printlabel@right{%
424     \ifx\eqL@SK@lab\relax\else
425       \rlap{\kern\marginparsep\eqL@SK@lab}%
426       \eqL@SK@clearlabel
427     \fi
428   }
429   \def\eqL@SK@printlabel@left{%
430     \ifx\eqL@SK@lab\relax\else
431       \llap{\eqL@SK@lab\kern\marginparsep}%
432       \eqL@SK@clearlabel
433     \fi
434   }
435   \def\eqL@SK@printlabel@line{%
436     \ifx\eqL@SK@lab\relax\else
437       \dimen@\prevdepth
438       \nointerlineskip
439       \ifdefined\eqL@tagsleft
440         \llap{%
441           \eqL@SK@lab
442           \kern\marginparsep
443         }%
444         \eqL@SK@clearlabel
445       \else
446         \rlap{%
447           \dimen@\displaywidth
448           \advance\dimen@\marginparsep
449           \kern\dimen@
450           \eqL@SK@lab
451         }%
452         \fi
453         \eqL@SK@clearlabel
454         \prevdepth\dimen@
455       \fi
456   }
457   \let\eqL@label@org\label
458   \def\eqL@label@clean{\let\SK@\@gobbletwo\eqL@label@org}
459 }

```

F.3 Labels

TODO: describe

```

460 % \TODO implement (via label[] or labelname similar to label/tag)
461 \let\eqL@nextlabel\@undefined

```

```
462 \def\eq@labelname@default{[equation]}
```

```
\eq@label@org
```

```
463 \let\eq@label@org\label
```

F.4 Tags

TODO: describe

```
464 \let\eq@nexttag\undefined
```

```
\eq@tag@default
```

```
465 \def\eq@tag@default{%
```

```
466 \eq@error{\string\tag\space not allowed here}{}\eq@tag@gobble}
```

```
467 \let\tag\eq@tag@default
```

```
\eq@tag@gobble TODO: ifnextchar, gobbletwo?
```

```
468 \def\eq@tag@gobble@[#1]#2{}
```

```
469 \def\eq@tag@gobble{%
```

```
470 \eq@ampprotecttwo\eq@teststaropt@tight\eq@tag@gobble@\eq@tag@gobble@{}}
```

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

```
\eq@tag@makenext
```

hyperref anchors

```
\eq@tag@makenext@@
```

```
\eq@tag@makenext@@@
```

```
471 \let\eq@Hy@anchor@gobble
```

```
472 \AddToHook{package/hyperref/after}{
```

```
473 \def\eq@Hy@anchor#1{%
```

```
474 \Hy@raisedlink{\hyper@anchor{#1}}%
```

```
475 }%
```

```
476 }
```

```
477 \def\eq@tag@makenext{%
```

```
478 \eq@ampprotecttwo\eq@teststaropt@tight
```

```
479 \eq@tag@makenext@star\eq@tag@makenext@\eq@tag@text
```

```
480 }
```

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

```
481 \def\eq@tag@makenext@star[#1]#2{%
```

```
482 \global\def\eq@nexttag{%
```

```
483 \let\eq@tag@tool\@firstofone
```

```
484 \protected@edef\eq@tag@text{#2}%
```

```
485 \protected@edef\eq@tag@label{#1}%
```

```
486 }%
```

```
487 }
```

```
488 \def\eq@tag@makenext@[#1]#2{%
```

```
489 \global\def\eq@nexttag{%
```

```
490 \let\eq@tag@tool\eq@tag@form
```

```
491 \protected@edef\eq@tag@text{#2}%
```

```
492 \protected@edef\eq@tag@label{#1}%
```

```
493 \protected@edef\eq@tag@label{\p@equation\eq@tag@label}%
```

```
494 }%
```

```
495 }
```

F.5 Anchors

TODO: describe

`g@refcount@ (counter)`

```
496 \newcount\eql@numbering@refcount@
497 \eql@numbering@refcount@z@
498 \def\eql@numbering@ref{equation.eql-\the\eql@numbering@refcount@}
499 \def\eql@numbering@refstep{\global\advance\eql@numbering@refcount@ \@ne}
```

TODO: describe

```
500 \def\eql@numbering@makeblockanchor{%
501   \eql@numbering@refstep
502   \global\edef\eql@label@currentHref{\eql@numbering@ref}%
503   \eql@Hy@anchor\eql@label@currentHref
504   \global\edef\eql@label@thepage{\thepage}%
505 }
506 \def\eql@numbering@setblockanchor{%
507   \let\thepage\eql@label@thepage
508   \let\@currentHref\eql@label@currentHref
509 }
```

F.6 Tag Composition

TODO: describe

```
\eql@compose@anchor
  \eql@compose@tag
  \eql@compose@label
510 \def\eql@compose@anchor{%
511   \ifdefined\eql@nexttag
512     \eql@nexttag
513     \def\@currentcounter{equation}%
514     \let\@currentlabel\eql@tag@label
515     \eql@numbering@refstep
516     \edef\@currentHref{\eql@numbering@ref}%
517     \eql@Hy@anchor\@currentHref
518     \global\let\eql@nexttag\@undefined
519   \else
520     \refstepcounter{equation}%
521     \let\eql@tag@tool\eql@tag@form
522     \edef\eql@tag@text{\theequation}%
523   \fi
524 }

525 \def\eql@compose@label{%
526   \ifmeasuring@\else
527     \eql@SK@clearlabel
528     \ifdefined\eql@nextlabel
529       \ifnum
530         \ifnum\eql@numbering@target@<\z@
531           \eql@row@
532         \else
533           \eql@numbering@target@
534         \fi=\eql@row@
535       \eql@compose@label@
536     \fi
```

```

537   \fi
538   \fi
539 }

```

TODO: describe

```

540 \def\eql@compose@label@{%
541   \eql@SK@setlabel\eql@nextlabel
542   \begingroup
543     \ifnum\eql@numbering@target@=\eql@row@
544       \eql@numbering@setblockanchor
545     \fi
546     \let\@currentlabelname\eql@labelname@default
547     \expandafter\eql@label@clean\expandafter{\eql@nextlabel}%
548     \global\let\eql@nextlabel\@undefined
549   \endgroup
550 }

```

TODO: describe

```

551 \def\eql@compose@tag{%
552   \eql@tagging@tagbegin
553   \eql@tag@box{%
554     \eql@tag@tool\eql@tag@text
555     \eql@tagging@tagsave
556   }%
557   \eql@tagging@tagend
558 }

```

TODO: describe

```

559 \def\eql@compose@print{%
560   \eql@compose@anchor
561   \eql@compose@label
562   \ifdefined\eql@tagsleft
563     \eql@SK@printlabel@left
564     \eql@compose@tag
565   \else
566     \eql@compose@tag
567     \eql@SK@printlabel@right
568   \fi
569 }

```

TODO: describe

```

570 \def\eql@compose@measure{%
571   \ifdefined\eql@nexttag
572     \eql@nexttag
573     \eql@tag@box{\eql@tag@tool\eql@tag@text}%
574   \else
575     \stepcounter{equation}%
576     \eql@tag@box{\eql@tag@form\theequation}%
577   \fi
578   \ifnum\eql@numbering@target@<\z@
579     \global\let\eql@nextlabel\@undefined
580     \global\let\eql@nexttag\@undefined
581   \fi
582 }

```


F.7 Tagbox Methods

TODO: describe

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

```
583 \def\eq@tagbox@make#1{%
584   \setbox\eq@tagbox@\hbox{\eq@strut@tag@\lign#1}%
585   \eq@tagwidth@\wd\eq@tagbox@
586   \ifdim\eq@tagwidth@<\eq@tagwidthmin@
587     \eq@tagwidth@\eq@tagwidthmin@
588   \fi
589   \advance\eq@tagwidth@\eq@tagsepmin@
590 }
```

TODO: describe

```
591 \def\eq@tagbox@print@tagsright{%
592   \kern-\wd\eq@tagbox@
593   \box\eq@tagbox@
594 }
```

TODO: describe

```
595 \def\eq@tagbox@print@tagsleft{%
596   \wd\eq@tagbox@\z@
597   \box\eq@tagbox@
598 }
```

TODO: describe

```
599 \def\eq@tagbox@print@tagsright@raise{%
600   \ifnum\eq@row@=\eq@totalrows@
601     \global\advance\eq@raisetag@firstlast@\tw@
602   \fi
603   \llap{\vtop{%
604     \vskip-\eq@raisetag@amount@
605     \normalbaselines
606     \setbox\@ne\hbox{%
607       \dp\@ne\eq@line@depth@
608       \box\@ne
609       \box\eq@tagbox@
610     }}%
611 }
612 \def\eq@tagbox@print@tagsleft@raise{%
613   \ifnum\eq@row@=\@one
614     \global\advance\eq@raisetag@firstlast@\@one
615   \fi
616   \rlap{\vbox{%
617     \normalbaselines
618     \box\eq@tagbox@
619     \vbox to\eq@line@height@{%
620       \vskip\eq@raisetag@amount@
621     }}%
622 }
```

TODO: describe

```
623 \def\eq@numbering@printsubeqlabel{%
624   \ifdefined\eq@parentlabel
625     \eq@numbering@makeblockanchor
626     \eq@SK@setlabel\eq@parentlabel
```

```

627 \begingroup
628   \def\@currentcounter{equation}%
629   \eql@numbering@setblockanchor
630   \let\@currentlabelname\eql@labelname@default
631   \protected@edef\@currentlabel{\p@equation\theparentequation}%
632   \expandafter\eql@label@clean\expandafter{\eql@parentlabel}%
633 \endgroup
634 \eql@SK@printlabel@line
635 \fi
636 }

```

F.8 Numbering Schemes

TODO: describe

```

637 \def\eql@numbering@tab@first{first}
638 \def\eql@numbering@tab@last{last}
639 \def\eql@numbering@tab@middle{middle}
640 \def\eql@numbering@tab@here{here}
641 \def\eql@numbering@tab@in{in}
642 \def\eql@numbering@tab@out{out}
643 \def\eql@numbering@tab@sub{sub}
644 \def\eql@numbering@tab@all{all}
645 \def\eql@numbering@tab@none{none}

```

TODO: describe

```

646 \let\eql@numbering@tab@f\eql@numbering@tab@first
647 \let\eql@numbering@tab@l\eql@numbering@tab@last
648 \let\eql@numbering@tab@m\eql@numbering@tab@middle
649 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
650 \let\eql@numbering@tab@o\eql@numbering@tab@out
651 \let\eql@numbering@tab@outside\eql@numbering@tab@out
652 \let\eql@numbering@tab@i\eql@numbering@tab@in
653 \let\eql@numbering@tab@inside\eql@numbering@tab@in
654 \let\eql@numbering@tab@within\eql@numbering@tab@in
655 \let\eql@numbering@tab@h\eql@numbering@tab@here
656 \let\eql@numbering@tab@s\eql@numbering@tab@sub
657 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
658 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
659 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
660 \let\eql@numbering@tab@a\eql@numbering@tab@all
661 \let\eql@numbering@tab@n\eql@numbering@tab@none
662 \expandafter\let\csname eql@numbering@tab@!\endcsname\eql@numbering@tab@all
663 \expandafter\let\csname eql@numbering@tab@*\endcsname\eql@numbering@tab@none
664 \expandafter\let\csname eql@numbering@tab@1\endcsname\eql@numbering@tab@first
665 \let\eql@numbering@mode\eql@numbering@tab@all
666 \def\eql@numbering@set#1{%
667   \ifcsname eql@numbering@tab@#1\endcsname
668     \expandafter\let\expandafter\eql@numbering@mode
669     \csname eql@numbering@tab@#1\endcsname
670     \ifx\eql@numbering@mode\eql@numbering@tab@none
671       \let\eql@numbering@mode\eql@numbering@tab@all
672       \let\eql@numbering@active\eql@false
673   \fi
674 \else
675   \eql@error{numbering mode ‘#1’ unknown: setting to ‘all’}%

```

```

676 \let\eq@numbering@mode\eq@numbering@tab@all
677 \fi
678 }

```

ring@target@ (*counter*)

```

679 \let\eq@numbering@active\eq@true
680 \newcount\eq@numbering@target@

681 \def\eq@numbering@mode@all{%
682 \eq@numbering@target@\m@ne}
683 \def\eq@numbering@mode@sub{%
684 \eq@numbering@target@\m@ne
685 \let\eq@numbering@subeq@use\eq@true}
686 \def\eq@numbering@mode@first{%
687 \eq@numbering@target@\@ne}
688 \def\eq@numbering@mode@last{%
689 \eq@numbering@target@\@MM}
690 \def\eq@numbering@mode@here{%
691 \eq@numbering@target@\z@}

```

TODO: describe

```

692 \def\eq@numbering@mode@in{%
693 \ifdefined\eq@tagsleft
694 \eq@numbering@mode@last
695 \else
696 \eq@numbering@mode@first
697 \fi
698 }

```

TODO: describe

```

699 \def\eq@numbering@mode@out{%
700 \ifdefined\eq@tagsleft
701 \eq@numbering@mode@first
702 \else
703 \eq@numbering@mode@last
704 \fi
705 }

```

TODO: describe

```

706 \def\eq@numbering@mode@middle{%
707 \eq@numbering@target@\z@
708 \let\eq@numbering@eval@target\eq@numbering@eval@middle}
709 \def\eq@numbering@eval@middle{%
710 \ifnum\eq@numbering@target@=\z@
711 \count@\eq@row@
712 \advance\count@\@ne
713 \divide\count@\tw@
714 \global\eq@numbering@target@\count@
715 \fi
716 }

```

TODO: describe

```

717 \def\eq@numbering@eval@mode{%
718 \let\eq@numbering@eval@target\@undefined
719 \let\eq@numbering@subeq@use\eq@false
720 \csname eq@numbering@mode@\eq@numbering@mode\endcsname

```

```

721 \ifdefined\eql@numbering@active
722   \let\eql@numbering@eqnswinit\@eqnswtrue
723 \else
724   \let\eql@numbering@eqnswinit\@eqnswfalse
725 \fi
726 \let\eql@numbering@active\eql@false
727 }

```

TODO: reconsider operation

`\numberhere`

```

728 \def\numberhere{%
729   \ifnum\eql@numbering@target@<\z@
730     \donumber
731   \else
732     \ifmeasuring@
733       \global\eql@numbering@target@\eql@row@
734     \fi
735   \fi
736 }

```

TODO: describe

`\numbernext`

```

737 \def\numbernext{%
738   \ifnum\eql@numbering@target@<\z@
739     \nonumber
740   \else
741     \ifmeasuring@
742       \ifnum\eql@numbering@target@=\eql@row@
743         \global\advance\eql@numbering@target@\@ne
744       \fi
745     \fi
746   \fi
747 }

```

F.9 Numbering Framework

TODO: describe

```

748 \let\eql@numbering@autolabel\eql@false
749 \let\eql@numbering@autotag\eql@true
750 \let\eql@numbering@blocklabel\@undefined
751 \let\eql@numbering@blocktag\@undefined
752 \let\eql@numbering@warn\eql@true

753 \def\eql@numbering@autoenable{%
754   \global\@eqnswtrue
755   \ifx\eql@numbering@mode\eql@numbering@tab@here
756     \ifnum\eql@numbering@target@=\z@
757       \numberhere
758     \fi
759   \fi
760 }

761 \eql@amsmath@after{
762   \let\eql@print@eqnum@default\print@eqnum

```

```
763 \let\eq@incr@eqnum@default\incr@eqnum
764 }
```

TODO: describe

```
765 \def\donumber{%
766   \if@eqnsw\else
767     \global\@eqnswtrue
768     \ifx\print@eqn\@empty
769       \global\let\print@eqn\eq@print@eqnum@default
770     \fi
771     \ifx\incr@eqn\@empty
772       \global\let\incr@eqn\eq@incr@eqnum@default
773     \fi
774   \fi
775 }
```

TODO: describe

```
776 \def\eq@label{%
777   \ifdefined\eq@numbering@autolabel
778     \eq@numbering@autoenable
779   \fi
780   \ifdefined\eq@numbering@warn
781     \ifdefined\eq@nextlabel
782       \eq@warn@label@multiple\eq@nextlabel
783     \fi
784   \fi
785   \global\edef\eq@nextlabel
786 }
```

TODO: describe

```
787 \let\eq@label@gobble\@gobble
```

TODO: describe

```
788 \def\eq@tag{%
789   \ifdefined\eq@numbering@autotag
790     \eq@numbering@autoenable
791   \fi
792   \ifdefined\eq@numbering@warn
793     \ifdefined\eq@nexttag
794       \eq@warn@tag@multiple
795     \fi
796   \fi
797   \eq@tag@makenext
798 }
```

TODO: describe

```
799 \def\eq@blocklabel@set#1{%
800   \ifdefined\eq@blocklabel
801     \eq@warn@label@multiple\eq@blocklabel
802   \fi
803   \edef\eq@blocklabel{#1}%
804 }
```

TODO: describe

```
805 \def\eq@blocktag@set#1{%
806   \ifdefined\eq@blocktag
```

```

807 \eql@warn@tag@multiple
808 \fi
809 \def\eql@blocktag{#{1}}%
810 }

```

TODO: describe

```

811 \def\eql@blocktag@setstar#1{%
812 \ifdefined\eql@blocktag
813 \eql@warn@tag@multiple
814 \fi
815 \def\eql@blocktag{*{#1}}%
816 }

```

Single Line. **TODO:** describe

```

817 \def\eql@numbering@single@init{%
818 \let\eql@numbering@warn\eql@true
819 \let\label\eql@label
820 \let\tag\eql@tag
821 \let\raisetag\eql@raisetag
822 \eql@numbering@target@{m@ne
823 \let\eql@nextlabel\eql@blocklabel
824 \ifdefined\eql@blocktag
825 \expandafter\eql@tag@makenext\eql@blocktag
826 \else
827 \let\eql@nexttag@undefined
828 \fi
829 \eql@numbering@eqnswinit
830 \ifdefined\eql@numbering@autolabel
831 \ifdefined\eql@nextlabel
832 \@eqnswtrue
833 \fi
834 \fi
835 \ifdefined\eql@numbering@autotag
836 \ifdefined\eql@nexttag
837 \@eqnswtrue
838 \fi
839 \fi
840 \global\eql@raisetag@amount@{z@
841 }

```

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

```

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

```

```

856     \ifdefined\eql@nextlabel
857     \@eqnswtrue
858     \fi
859     \fi
860     \fi
861 }

```

TODO: describe

```

862 \def\eql@numbering@measure@line@begin{%
863   \ifnum\eql@numbering@target@<\z@
864     \global\eql@numbering@eqnswinit
865   \fi
866 }

```

TODO: describe

```

867 \def\eql@numbering@measure@eval{%
868   \ifdefined\eql@numbering@eval@target
869     \eql@numbering@eval@target
870   \fi
871   \ifnum\eql@numbering@target@>\eql@row@
872     \global\eql@numbering@target@\eql@row@
873   \fi
874   \ifnum\eql@numbering@target@>\z@
875     \if@eqnsw\else
876       \global\eql@numbering@target@\z@
877     \fi
878   \fi
879   \ifnum\eql@numbering@target@<\@ne
880     \ifdefined\eql@nextlabel
881       \eql@warn@label@unused
882       \global\let\eql@nextlabel\@undefined
883     \fi
884     \ifdefined\eql@nexttag
885       \eql@warn@tag@unused
886       \global\let\eql@nexttag\@undefined
887     \fi
888   \fi
889 }

```

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

```

890 \def\eql@numbering@print@init{%
891   \ifnum\eql@numbering@target@<\z@
892     \let\eql@numbering@warn\eql@false
893     \let\label\eql@label
894     \let\tag\eql@tag
895     \let\raisetag\eql@raisetag
896     \let\eql@nextlabel\eql@blocklabel
897     \ifdefined\eql@blocktag
898       \expandafter\eql@tag@makenext\eql@blocktag
899     \else
900       \let\eql@nexttag\@undefined
901     \fi
902   \else
903     \let\label\eql@label@gobble
904     \let\tag\eql@tag@gobble
905     \let\raisetag\eql@raisetag@gobble

```

```
906 \fi
907 }
```

TODO: describe

```
908 \def\eq@numbering@print@block@begin{%
909 \ifnum\eq@numbering@target@>\z@
910 \eq@numbering@makeblockanchor
911 \fi
912 \ifdefined\eq@numbering@subeq@use
913 \eq@numbering@printsubeqlabel
914 \fi
915 }
```

TODO: describe

```
916 \def\eq@numbering@print@line@begin{%
917 \ifnum\eq@numbering@target@<\z@
918 \global\eq@numbering@eqnswinit
919 \global\eq@raisetag@amount@>\z@
920 \fi
921 }
```

TODO: describe

```
922 \def\eq@numbering@print@line@eval{%
923 \ifnum\eq@numbering@target@<\z@ \else
924 \ifnum\eq@numbering@target@=\eq@row@
925 \global\@eqnswtrue
926 \else
927 \global\@eqnswfalse
928 \fi
929 \fi
930 }
```

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:

```
931 \eq@amsmath@undefine\c@parentequation
932 \eq@amsmath@undefine\theparentequation
933 \ifdefined\c@parentequation\else
934 \newcounter{parentequation}
935 \fi
```

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

```
936 \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:

```
937 \let\eq@subequations@active\eq@false
```

`ql@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`:

```
938 \def\eq@subequations@init{%
939   \edef\eq@subequations@restorecounter{%
940     \global\c@equation\the\c@equation\relax}%
941   \ifdefined\eq@blocktag
942     \expandafter\eq@tag@makenext\eq@blocktag
943     \eq@nexttag
944     \eq@numbering@refstep
945     \protected@edef\theHparentequation{\eq@numbering@ref}%
946     \protected@edef\theparentequation{\eq@tag@text}%
947   \else
948     \advance\c@equation\@ne
949     \protected@edef\theparentequation{\theequation}%
950     \ifdefined\theHequation
951       \protected@edef\theHparentequation{\theHequation}%
952     \fi
953   \fi
954   \global\c@parentequation\c@equation
955   \global\c@equation\z@
956   \let\theequation\eq@subequations@template
957   \def\theHequation{\theHparentequation.\arabic{equation}}%
958 }
```

`l@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:

```
959 \def\eq@subequations@close{%
960   \ifnum\c@equation=\z@
961     \eq@subequations@restorecounter
962   \else
963     \global\c@equation\c@parentequation
964   \fi
965 }
```

G.2 Environment

`l@subequations@start` Start the subequations environment with optional parameters in #1. Enter subequations mode and set an anchor for subsequent `\label`'s. Manually print the showkeys tag:

TODO: join with other similar anchor routines `\eq@numbering@printsbeqlabel`

```
966 \def\eq@subequations@start{%
967   \let\eq@blocktag\@undefined
968   \let\eq@blocklabel\@undefined
969   \eq@nextopt@process{subequations}%
```

```

970 \eql@subequations@init
971 \eql@numbering@refstep
972 \edef\eql@subequations@currentHref{\eql@numbering@ref}%
973 \eql@Hy@anchor\eql@subequations@currentHref
974 \edef\eql@subequations@thepage{\thepage}%
975 \def\@currentcounter{equation}%
976 \let\@currentHref\eql@subequations@currentHref
977 \protected@edef\@currentlabel{\p@equation\theparentequation}%
978 \let\@currentlabelname\eql@labelname@default
979 \let\eql@subequations@active\eql@true
980 \ifdefined\eql@blocklabel
981   \eql@SK@label\eql@blocklabel
982 \fi
983 \ignorespaces
984 }

```

`\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:

```

985 \def\eql@subequations@end{%
986   \ifnum\c@equation>\z@
987     \ifdefined\eql@blocklabel
988       \begingroup
989         \def\@currentcounter{equation}%
990         \let\thepage\eql@subequations@thepage
991         \let\@currentHref\eql@subequations@currentHref
992 % \TODO how about tag* ?! also within equations!
993         \protected@edef\@currentlabel{\p@equation\theparentequation}%
994         \let\@currentlabelname\eql@labelname@default
995         \expandafter\eql@label@clean\expandafter{\eql@blocklabel}%
996       \endgroup
997       \fi
998     \fi
999   \eql@subequations@close
1000   \ignorespacesafterend
1001 }

```

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

```

1002 \newenvironment{eql@subequations}{%
1003 (dev)\eql@dev@enterenv
1004 \eql@subequations@testall\eql@subequations@start%
1005 }{%
1006 \eql@subequations@end
1007 (dev)\eql@dev@leaveenv
1008 }

```

TODO: describe

```

1009 \def\eql@subequations@testall{\eql@subequations@testopt}
1010 \def\eql@subequations@testopt#1{%
1011   \eql@ifnextchar@tight[%]
1012     {\eql@subequations@addopt{\eql@subequations@testat{#1}}}%
1013     {\eql@subequations@testat{#1}}}
1014 \def\eql@subequations@addopt#1[#2]{\eqnaddopt{#2}#1}
1015 \def\eql@subequations@testat#1{%
1016   \eql@ifat@tight%
1017     {\eql@subequations@addlabel{#1}}%

```

```

1018     {#1}}
1019 \def\eq@subequations@addlabel#1#2{\eqnaddopt{label={#2}}#1}

```

G.3 Subequation Scheme

TODO: describe

```

1020 \def\eq@numbering@subeq@init{%
1021   \let\eq@save@theequation\theequation
1022   \let\eq@save@theHequation\theHequation
1023   \eq@subequations@init
1024   \let\eq@parentlabel\eq@blocklabel
1025   \let\eq@parenttag\eq@blocktag
1026   \let\eq@blocklabel\@undefined
1027   \let\eq@blocktag\@undefined
1028 }

```

TODO: describe

```

1029 \def\eq@numbering@subeq@test{%
1030   \ifnum\c@equation<\tw@
1031     \let\eq@numbering@subeq@use\@ne
1032   \fi
1033 }

```

TODO: describe

```

1034 \def\eq@numbering@subeq@revert{%
1035   \let\eq@blocklabel\eq@parentlabel
1036   \let\eq@blocktag\eq@parenttag
1037   \let\eq@numbering@subeq@use\eq@false
1038   \let\theequation\eq@save@theequation
1039   \let\theHequation\eq@save@theHequation
1040   \eq@subequations@restorecounter
1041 }

```

TODO: describe

```

1042 % \TODO note must not use setcounter here (when calc is loaded)
1043 \def\eq@numbering@subeq@close{%
1044   \eq@subequations@close
1045 }

```

H Display Equations Support

TODO: describe

H.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```

1046 \interdisplaylinepenalty\@M

```

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

```

1047 \def\eql@getdsp@pen#1{%
1048   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1049 }

```

TODO: allow a displaybreak before equations

```

1050 \DeclareRobustCommand{\eql@displaybreak@default}[1][4]{%
1051   \eql@warning{Invalid use of \string\displaybreak}{}}
1052 \eql@amsmath@after{\let\eql@displaybreak@default\displaybreak}
1053 \eql@amsmath@let\displaybreak\eql@displaybreak@default

1054 \newcount\eql@displaybreak@pen@
1055 \newcount\eql@displaybreak@prepen@

1056 \protected\def\eql@displaybreak@print{%
1057   \eql@ampprotect\eql@testopt@tight\eql@displaybreak@print@{4}%
1058 }

```

TODO: describe

```

1059 \def\eql@displaybreak@print@[#1]{%
1060   \ifnum#1<\z@
1061     \global\eql@displaybreak@pen@\@MM
1062   \else
1063     \global\eql@displaybreak@pen@-\@getpen{#1}\relax
1064   \fi
1065 }

```

TODO: describe

```

1066 \def\eql@displaybreak@pre#1{%
1067   \ifnum#1<\z@
1068     \global\eql@displaybreak@prepen@\@MM
1069   \else
1070     \global\eql@displaybreak@prepen@-\@getpen{#1}\relax
1071   \fi
1072 }

```

TODO: describe

```

1073 \protected\def\eql@displaybreak@measure{%
1074   \eql@ampprotect\eql@testopt@tight\eql@displaybreak@measure@{4}%
1075 }
1076 \def\eql@displaybreak@measure@[#1]{%

```

H.2 General Initialisation

TODO: describe

```

\eql@vspaceskip@ (skip) TODO: add a proper star variant?!
\eql@abovespace@ (skip)
\eql@belowspace@ (skip)
1077 \newskip\eql@vspaceskip@
1078 \newskip\eql@abovespace@
1079 \newskip\eql@belowspace@
1080 \let\eql@vspace@org\vspace
1081 \def\eql@vspace@{\eql@ifstar@loose\eql@vspace@\eql@vspace@}
1082 \def\eql@vspace@#1{%
1083   \global\advance\eql@vspaceskip@\glueexpr#1\relax}

```

\eql@display@init

```

1084 \def\eqldisplay@init{%
1085   \eqldisplay@firstavail@z@
1086   \eql@raisetag@firstlast@z@
1087   \let\displaybreak\eqldisplaybreak@print
1088   \eqldisplaybreak@open@MM
1089   \eqlvspaceskip@z@skip
1090   \let\eqlvspace@org\vspace
1091   \let\vspace\eqlvspace
1092 }

```

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

```

1093 \def\eqldisplay@close{%
1094 % \TODO temporary fix for development stages
1095   \ifdefined\eql@tagging@on
1096     \ifdefined\dollardollar@begin\else
1097       \belowdisplayskip-\belowdisplayskip
1098       \belowdisplayshortskip-\belowdisplayshortskip
1099     \fi
1100   \fi
1101   \ifdim\eqldisplay@firstavail@<z@
1102     \eqldisplay@firstavail@z@
1103   \fi
1104   \eql@skip@mode@leave@z@
1105   \ifdim\eql@halign@prevdepth@=\maxdimen
1106     \ifdim\predisplaysize=-\maxdimen
1107       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1108       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1109     \else
1110       \eql@skip@mode@above@z@
1111       \eql@skip@mode@below@z@
1112       \advance\eqldisplay@firstavail@\displayindent
1113       \ifdim\eqldisplay@firstavail@>\predisplaysize
1114         \ifcase\eql@skip@mode@short\relax
1115         \or
1116           \eql@skip@mode@above@\@ne
1117         \or
1118           \eql@skip@mode@above@\@ne
1119           \ifnum\eql@totalrows@=\@ne
1120             \eql@skip@mode@below@\@ne
1121           \fi
1122         \or
1123           \eql@skip@mode@above@\@ne
1124           \eql@skip@mode@below@\@ne
1125         \fi
1126       \fi
1127     \fi
1128   \else
1129     \ifdim\eql@halign@prevdepth@=-\@m\p@
1130       \eql@skip@mode@above@\eql@skip@mode@top@above\relax
1131       \eql@skip@mode@below@\eql@skip@mode@top@below\relax
1132     \else
1133       \eql@skip@mode@above@\eql@skip@mode@par@above\relax
1134       \eql@skip@mode@below@\eql@skip@mode@par@below\relax
1135     \fi
1136   \fi
1137   \ifcase\eql@skip@mode@above@

```

```

1138 \or\or\or
1139 \predisplaypenalty\z@
1140 \or
1141 \predisplaypenalty\z@
1142 \fi
1143 \ifcase\eq\skip@mode@below@
1144 \or\or\or
1145 \eq\skip@mode@leave@\@ne
1146 \or
1147 \eq\skip@mode@leave@\tw@
1148 \fi
1149 \ifdefined\eq\skip@force@above
1150 \eq\skip@mode@above@\eq\skip@force@above\relax
1151 \fi
1152 \ifdefined\eq\skip@force@below
1153 \eq\skip@mode@below@\eq\skip@force@below\relax
1154 \fi
1155 \ifdefined\eq\skip@force@leave
1156 \eq\skip@mode@leave@\eq\skip@force@leave\relax
1157 \fi
1158 \ifnum\eq\skip@mode@leave@>\z@
1159 \postdisplaypenalty\z@
1160 \fi
1161 \ifodd\eq\raisetag@firstlast@
1162 \ifcase\eq\skip@mode@above@
1163 \abovedisplayskip\glueexpr\eq\skip@tag@above\relax
1164 \or
1165 \abovedisplayskip\glueexpr\eq\skip@tag@above\relax
1166 \or
1167 \abovedisplayskip\glueexpr\eq\skip@tag@above\relax
1168 \or
1169 \abovedisplayskip\glueexpr\eq\skip@partag@above\relax
1170 \or
1171 \abovedisplayskip\glueexpr\eq\skip@partag@above\relax
1172 \or
1173 \abovedisplayskip\z@skip
1174 \or
1175 \abovedisplayskip\glueexpr\eq\skip@medtag@above\relax
1176 \or
1177 \abovedisplayskip\glueexpr\eq\skip@custom@above\relax
1178 \fi
1179 \else
1180 \ifcase\eq\skip@mode@above@
1181 \abovedisplayskip\glueexpr\eq\skip@long@above\relax
1182 \or
1183 \abovedisplayskip\glueexpr\eq\skip@short@above\relax
1184 \or
1185 \abovedisplayskip\glueexpr\eq\skip@cont@above\relax
1186 \or
1187 \abovedisplayskip\glueexpr\eq\skip@par@above\relax
1188 \or
1189 \abovedisplayskip\glueexpr\eq\skip@top@above\relax
1190 \or
1191 \abovedisplayskip\z@skip
1192 \or
1193 \abovedisplayskip\glueexpr\eq\skip@med@above\relax
1194 \or
1195 \abovedisplayskip\glueexpr\eq\skip@custom@above\relax

```

```

1196   \fi
1197 \fi
1198 \ifnum\eq@raisetag@firstlast@>\@ne
1199   \ifcase\eq@skip@mode@below@
1200     \belowdisplayskip\glueexpr\eq@skip@tag@below\relax
1201   \or
1202     \belowdisplayskip\glueexpr\eq@skip@tag@below\relax
1203   \or
1204     \belowdisplayskip\glueexpr\eq@skip@tag@below\relax
1205   \or
1206     \belowdisplayskip\glueexpr\eq@skip@partag@below\relax
1207   \or
1208     \belowdisplayskip\glueexpr\eq@skip@partag@below\relax
1209   \or
1210     \belowdisplayskip\z@skip
1211   \or
1212     \belowdisplayskip\glueexpr\eq@skip@medtag@below\relax
1213   \or
1214     \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
1215   \fi
1216 \else
1217   \ifcase\eq@skip@mode@below@
1218     \belowdisplayskip\glueexpr\eq@skip@long@below\relax
1219   \or
1220     \belowdisplayskip\glueexpr\eq@skip@short@below\relax
1221   \or
1222     \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
1223   \or
1224     \belowdisplayskip\glueexpr\eq@skip@par@below\relax
1225   \or
1226     \belowdisplayskip\glueexpr\eq@skip@top@below\relax
1227   \or
1228     \belowdisplayskip\z@skip
1229   \or
1230     \belowdisplayskip\glueexpr\eq@skip@med@below\relax
1231   \or
1232     \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
1233   \fi
1234 \fi
1235 \ifnum\eq@displaybreak@open@=\@MM\else
1236   \postdisplaypenalty\eq@displaybreak@open@
1237 \fi
1238 \ifnum\eq@displaybreak@prepen@=\@MM\else
1239   \predisplaypenalty\eq@displaybreak@prepen@
1240 \fi
1241 \advance\abovedisplayskip\eq@abovespace@
1242 \advance\belowdisplayskip\eq@belowspace@
1243 \advance\belowdisplayskip\eq@vspaceskip@
1244 \count@\prevgraf
1245 \advance\count@\eq@totalrows@
1246 \ifnum\count@>\z@
1247   \advance\count@\m@ne
1248 \fi
1249 \prevgraf\count@
1250 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
1251 % \TODO temporary fix for development stages
1252 \ifdefined\eq@tagging@on
1253   \ifdefined\dollardollar@begin\else

```

```

1254     \belowdisplayskip-\belowdisplayskip
1255     \fi
1256 \fi
1257 }

1258 \def\eqldisplay@leave{%
1259   \let\label\eqldisplay@label@org
1260   \let\tag\eqldisplay@tag@default
1261   \let\raisetag\eqldisplay@raisetag@default
1262   \let\displaybreak\eqldisplay@displaybreak@default
1263   \let\intertext\eqldisplay@intertext@default
1264   \let\vspace\eqldisplay@vspace@org
1265 }
1266 \expandafter\def\expandafter\@arrayparboxrestore\expandafter{%
1267   \@arrayparboxrestore
1268   \eqldisplay@leave
1269   \ifdefined\eqldisplay@ampproof@active
1270     \eqldisplay@ampprevert
1271   \fi
1272   \@displayfalse
1273 }

```

H.3 halign Support

TODO: describe

`\eqldisplay@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.

```

1274 \newbox\eqldisplay@strutbox@
1275 \def\eqldisplay@strut{\copy\eqldisplay@strutbox@}
1276 \let\eqldisplay@strut@field\eqldisplay@strut
1277 \let\eqldisplay@strut@tag\eqldisplay@strut
1278 \def\eqldisplay@strut@make{%
1279   \setbox\eqldisplay@strutbox@\hbox{%
1280     \@tempdima\normalbaselineskip
1281     \advance\@tempdima-\normallineskiplimit
1282     \@tempdimb.3\normalbaselineskip
1283     \advance\@tempdimb.5\normallineskiplimit
1284     \advance\@tempdima-\@tempdimb
1285     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1286   }
1287 }
1288 \AtBeginDocument{\eqldisplay@strut@make}

```

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

```

1289 \def\eqldisplay@halign@spread{%
1290   \eqldisplay@spread@\glueexpr\eqldisplay@spread@val\relax
1291   \advance\eqldisplay@spread@\normalbaselineskip
1292   \advance\eqldisplay@spread@-\baselineskip
1293   \ifdim\eqldisplay@spread@>\z@
1294     \openup\eqldisplay@spread@
1295     \ifdefined\spread@equation
1296       \let\spread@equation\@empty
1297     \fi
1298   \fi
1299 }

```


gn@prevdepth@ (*dimen*)

```
1300 \newdimen\eql@halign@prevdepth@
1301 \def\eql@halign@catchprevdepth{%
1302   \ifvmode
1303     \eql@halign@prevdepth@\prevdepth
1304     \nointerlineskip
1305     \noindent
1306   \else
1307     \eql@halign@prevdepth@\maxdimen
1308   \fi
1309 }
```

```
1310 \def\eql@halign@leave{%
1311   \ifcase\eql@skip@mode@leave@
1312   \or
1313     \endgraf
1314   \or
1315     \endgraf
1316     \prevdepth-\@m\p@
1317   \fi
1318 }
```

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

```
1319 \def\eql@halign@before{%
1320   \ifdim\eql@halign@prevdepth@=\maxdimen\else
1321     \prevdepth\eql@halign@prevdepth@
1322   \fi
1323   \ifdim\prevdepth=-\@m\p@\else
1324     \ifdefined\eql@display@height
1325       \skip@\baselineskip
1326       \advance\skip@-\glueexpr\eql@display@height\relax
1327       \advance\skip@-\prevdepth\relax
1328       \ifdim\skip@<\lineskiplimit
1329         \skip@\lineskip
1330       \fi
1331       \advance\skip@-\eql@spread@\relax
1332       \vskip\skip@
1333       \nointerlineskip
1334     \else
1335       \vskip-\eql@spread@\relax
1336     \fi
1337   \fi
1338 }
```

TODO: describe

```
1339 \def\eql@halign@after{%
1340   \ifdefined\eql@display@depth
1341     \prevdepth\glueexpr\eql@display@depth\relax
1342   \fi
1343 }
```

TODO: describe

```
1344 \def\eql@halign@init#1{%
1345   \eql@halign@spread
1346   \eql@strut@make
1347   \everycr{\noalign{#1}}%
1348 }
```

H.4 Stack

TODO: describe

```
1349 \def\eqL@stack@enable{%
1350   \let\eqL@stack@save@single\eqL@stack@save@single@
1351   \let\eqL@stack@save@multi\eqL@stack@save@multi@
1352   \let\eqL@stack@save@boxed\eqL@stack@save@boxed@
1353 }
```

TODO: describe

```
1354 \let\eqL@stack@save@single\eqL@stack@enable
1355 \let\eqL@stack@save@multi\eqL@stack@enable
1356 \let\eqL@stack@save@boxed\eqL@stack@enable
1357 \let\eqL@stack@restore\@empty
```

TODO: describe

```
1358 \def\eqL@stack@save@reg#1{\global#1\the#1\relax}
1359 \def\eqL@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

TODO: describe

```
1360 \def\eqL@stack@save@single@{%
1361   \let\eqL@stack@nextlabel\eqL@nextlabel
1362   \let\eqL@stack@nexttag\eqL@nexttag
1363   \edef\eqL@stack@restore{%
1364     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1365     \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1366     \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1367     \eqL@stack@save@reg\eqL@displaybreak@pen@
1368     \eqL@stack@save@reg\eqL@vspaceskip@
1369     \eqL@stack@save@reg\eqL@shape@pos@
1370     \eqL@stack@save@reg\eqL@shape@amount@
1371     \eqL@stack@save@reg\eqL@display@firstavail@
1372     \eqL@stack@save@reg\eqL@raisetag@amount@
1373     \eqL@stack@save@reg\eqL@raisetag@firstlast@
1374   }%
1375 }
```

TODO: describe

```
1376 \def\eqL@stack@save@multi@{%
1377   \let\eqL@stack@nextlabel\eqL@nextlabel
1378   \let\eqL@stack@nexttag\eqL@nexttag
1379   \let\eqL@stack@tagwidth@tab\eqL@tagwidth@tab
1380   \let\eqL@stack@fieldlength@tab\eqL@fieldlength@tab
1381   \let\eqL@stack@colwidth@tab\eqL@colwidth@tab
1382   \let\eqL@stack@label@thepage\eqL@label@thepage
1383   \let\eqL@stack@label@currentHref\eqL@label@currentHref
1384   \edef\eqL@stack@restore{%
1385     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1386     \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1387     \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1388     \eqL@stack@save@let\eqL@stack@tagwidth@tab\eqL@tagwidth@tab
1389     \eqL@stack@save@let\eqL@stack@fieldlength@tab\eqL@fieldlength@tab
1390     \eqL@stack@save@let\eqL@stack@colwidth@tab\eqL@colwidth@tab
1391     \eqL@stack@save@let\eqL@stack@label@thepage\eqL@label@thepage
1392     \eqL@stack@save@let\eqL@stack@label@currentHref\eqL@label@currentHref
1393     \eqL@stack@save@reg\eqL@displaybreak@pen@
```

```

1394 \eql@stack@save@reg\eql@vspaceskip@
1395 \eql@stack@save@reg\eql@shape@pos@
1396 \eql@stack@save@reg\eql@shape@amount@
1397 \eql@stack@save@reg\eql@display@firstavail@
1398 \eql@stack@save@reg\eql@raisetag@amount@
1399 \eql@stack@save@reg\eql@raisetag@firstlast@
1400 \eql@stack@save@reg\eql@column@
1401 \eql@stack@save@reg\eql@totalcolumns@
1402 \eql@stack@save@reg\eql@line@avail@
1403 \eql@stack@save@reg\eql@line@pos@
1404 \eql@stack@save@reg\eql@line@width@
1405 \eql@stack@save@reg\eql@line@depth@
1406 \eql@stack@save@reg\eql@line@height@
1407 \eql@stack@save@reg\eql@tagwidth@max@
1408 \eql@stack@save@reg\eql@numbering@target@
1409 \eql@stack@save@reg\eql@row@
1410 \eql@stack@save@reg\eql@tagrows@
1411 }%
1412 }
1413 \def\eql@stack@save@boxed@{%
1414 \edef\eql@stack@restore@{%
1415 \eql@stack@save@reg\eql@row@
1416 \eql@stack@save@reg\eql@totalrows@
1417 \eql@stack@save@reg\eql@shape@pos@
1418 \eql@stack@save@reg\eql@shape@amount@
1419 }%
1420 }

```

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:

```

1421 \ifdefined\inf@bad\else%
1422 \newcount\inf@bad
1423 \inf@bad1000000\relax
1424 \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`:

```

1425 \let\eql@restore@hfuzz\@empty
1426 \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:

```

1427 \newcount\eql@shape@pos@
1428 \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:

```
1429 \newdimen\eql@marginleft@
1430 \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:

```
1431 \newcount\eql@marginbadness@
1432 \newcount\eql@maxbadness@
1433 \eql@marginbadness@\inf@bad
1434 \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:

```
1435 \def\eql@shape@tab@default{default}
1436 \def\eql@shape@tab@left{left}
1437 \def\eql@shape@tab@center{center}
1438 \def\eql@shape@tab@right{right}
1439 \def\eql@shape@tab@first{first}
1440 \def\eql@shape@tab@hanging{hanging}
1441 \def\eql@shape@tab@steps{steps}
```

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

```
1442 \let\eql@shape@tab@def\eql@shape@tab@default
1443 \let\eql@shape@tab@\eql@shape@tab@default
1444 \let\eql@shape@tab@l\eql@shape@tab@left
1445 \let\eql@shape@tab@c\eql@shape@tab@center
1446 \let\eql@shape@tab@r\eql@shape@tab@right
1447 \let\eql@shape@tab@rc\eql@shape@tab@first
1448 \let\eql@shape@tab@indent\eql@shape@tab@first
1449 \let\eql@shape@tab@hang\eql@shape@tab@hanging
1450 \let\eql@shape@tab@lc\eql@shape@tab@hanging
1451 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
1452 \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:

```
1453 \let\eql@shape@mode\eql@shape@tab@default
```

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

```
1454 \def\eql@shape@set#1{%
1455   \ifcsname eql@shape@tab@#1\endcsname
1456     \expandafter\let\expandafter\eql@shape@mode
1457     \csname eql@shape@tab@#1\endcsname
1458   \else
1459     \eql@error{shape ‘#1’ unknown: setting to default}%
1460     \let\eql@shape@mode\eql@shape@tab@default
1461   \fi
1462 }
```

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

```

1463 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
1464 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne@eq@shape@amount@z@}
1465 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
1466 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
1467 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
1468 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne@eq@shape@amount@z@}
1469 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
1470 \let\eq@shape@layoutleft@default\eq@shape@layoutleft@left

```

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

```

1471 \def\eq@shape@layoutcenter@first{%
1472   \eq@shape@pos@z@
1473   \eq@shape@amount@z@
1474   \ifnum\eq@totalrows@>\@ne
1475     \ifnum\eq@row@=\@ne
1476       \eq@shape@amount@\eq@indent@
1477     \fi
1478   \fi
1479 }
1480 \def\eq@shape@layoutleft@first{%
1481   \eq@shape@pos@z@
1482   \eq@shape@amount@z@
1483   \ifnum\eq@totalrows@>\@ne
1484     \ifnum\eq@row@=\@ne
1485       \eq@shape@amount@\eq@indent@
1486     \fi
1487   \fi
1488 }

```

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:

```

1489 \def\eq@shape@layoutcenter@hanging{%
1490   \eq@shape@pos@z@
1491   \eq@shape@amount@\eq@indent@
1492   \ifnum\eq@totalrows@>\@ne
1493     \ifnum\eq@row@=\@ne
1494       \eq@shape@amount@z@
1495     \fi
1496   \fi
1497 }
1498 \def\eq@shape@layoutleft@hanging{%
1499   \eq@shape@pos@z@
1500   \eq@shape@amount@z@
1501   \ifnum\eq@totalrows@>\@ne
1502     \ifnum\eq@row@=\@ne
1503       \eq@shape@amount@-\eq@indent@
1504     \fi
1505   \fi
1506 }

```

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:

```

1507 \def\eq@shape@layoutcenter@steps{%
1508   \eq@shape@amount@z@
1509   \eq@shape@pos@\@ne
1510   \ifnum\eq@totalrows@>\@ne
1511     \ifnum\eq@row@=\@ne
1512       \eq@shape@pos@z@
1513     \fi
1514     \ifnum\eq@row@=\eq@totalrows@
1515       \eq@shape@pos@tw@
1516     \fi
1517   \fi
1518 }
1519 \def\eq@shape@layoutleft@steps{%
1520   \eq@shape@pos@z@
1521   \eq@shape@amount@z@
1522   \ifnum\eq@totalrows@>\@ne
1523     \ifnum\eq@row@=\@ne
1524       \eq@shape@amount@-\eq@indent@
1525     \fi
1526     \ifnum\eq@row@=\eq@totalrows@
1527       \eq@shape@amount@\eq@indent@
1528     \fi
1529   \fi
1530 }

```

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

```

1531 \let\eq@shape@eval\undefined
1532 \def\eq@shape@sel{%
1533   \expandafter\let\expandafter\eq@shape@eval
1534   \csname eq@shape%
1535     @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
1536     @\eq@shape@mode\endcsname
1537 }

```

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

```

1538 \def\eq@adjust@alignleft{%
1539   \global\eq@shape@pos@z@
1540   \eq@srbgroup\eq@ifstar@tight
1541   {\eq@adjust@alignleft@[\eq@indent@]}%
1542   {\eq@ifnextgobble@tight!}%
1543   {\eq@adjust@alignleft@[-\eq@indent@]}%
1544   {\eq@testopt@tight\eq@adjust@alignleft@z@}%
1545   }%
1546 }
1547 \def\eq@adjust@alignleft@[#1]{%
1548   \eq@sregroup\global\eq@shape@amount@\glueexpr#1\relax}
1549 \def\eq@adjust@aligncenter{%
1550   \global\eq@shape@pos@\@ne\global\eq@shape@amount@z@}
1551 \def\eq@adjust@alignright{%
1552   \global\eq@shape@pos@tw@\global\eq@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@`:

```
1553 \def\eql@adjust@try#1{%
1554   \setbox\eql@fieldbox@\hbox to\displaywidth{%
1555     \unhbox\eql@fieldbox@\unkern\kern#1}%
1556 }
```

`\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`:

```
1557 \def\eql@adjust@print#1#2{%
1558   \eql@restore@hfuzz
1559   \hbox to\displaywidth{%
1560     #1%
1561     \unhbox\eql@fieldbox@\unkern
1562     #2%
1563     \eql@tagging@mathaddlast
1564   }%
1565 }
```

`\eql@adjust@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:

```
1566 \def\eql@adjust@print@alignleft#1#2{%
1567   \ifnum\eql@row@=\@one
1568     \global\eql@display@firstavail@#1%
1569   \fi
1570   \eql@adjust@print{\kern#1}{\kern#2}%
1571 }
1572 \def\eql@adjust@print@alignright#1#2{%
1573   \ifnum\eql@row@=\@one
1574     \eql@display@firstavail@\displaywidth
1575     \advance\eql@display@firstavail@-\eql@fieldwidth@
1576     \global\advance\eql@display@firstavail@-#2%
1577   \fi
1578   \eql@adjust@print{\kern#1\hfil}{\unskip\kern#2}%
1579 }
1580 \def\eql@adjust@print@aligncenter#1#2{%
1581   \ifnum\eql@row@=\@one
1582     \eql@display@firstavail@\displaywidth
1583     \advance\eql@display@firstavail@-\eql@fieldwidth@
1584     \advance\eql@display@firstavail@#1%
1585     \advance\eql@display@firstavail@-#2%
1586     \global\divide\eql@display@firstavail@\tw@
1587   \fi
1588   \eql@adjust@print{\kern#1\hfil}{\kern#2}%
1589 }
```

`\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:

```
1590 \def\eql@adjust@init{%
1591   \eql@save@hfuzz
```

```

1592 \hfuzz\maxdimen
1593 \eql@shape@sel
1594 }

```

`\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:

```

1595 \def\eql@adjust@sel@tag{%
1596 \eql@tagging@tagaddbox
1597 \ifcase\eql@shape@pos@
1598 \eql@tagging@alignleft
1599 \or
1600 \eql@tagging@aligncenter
1601 \or
1602 \eql@tagging@alignright
1603 \fi
1604 \csname eql@adjust%
1605 @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
1606 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1607 @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
1608 @tag\endcsname
1609 }
1610 \def\eql@adjust@sel@notag{%
1611 \eql@tagging@tagaddbox
1612 \ifcase\eql@shape@pos@
1613 \eql@tagging@alignleft
1614 \or
1615 \eql@tagging@aligncenter
1616 \or
1617 \eql@tagging@alignright
1618 \fi
1619 \csname eql@adjust%
1620 @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
1621 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1622 @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
1623 @notag\endcsname
1624 }

```

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

```

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

```



```

1644     \eql@marginright@\dimen@
1645     \advance\eql@marginright@-\glueexpr\eql@paddingright@val\relax
1646     \ifdim\eql@marginright@<\z@
1647         \eql@marginright@\z@
1648     \fi
1649 \fi
1650 \fi
1651 }

```

I.4 Central Alignment Layout

TODO: describe

TODO: check all these!!

```

1652 \def\eql@adjust@layoutcenter@tagsright@aligncenter@notag{%
1653     \dimen@\displaywidth
1654     \advance\dimen@-\eql@fieldwidth@
1655     \ifdim\dimen@>\eql@tagmargin@
1656         \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1657     \else
1658         \eql@adjust@print@alignleft\z@\z@
1659     \fi
1660 }

```

TODO: describe

```

1661 \def\eql@adjust@layoutcenter@tagsright@aligncenter@tag{%
1662     \dimen@\displaywidth
1663     \ifdim\eql@tagwidth@<\eql@tagmargin@
1664         \advance\dimen@-\eql@tagmargin@
1665     \else
1666         \advance\dimen@-2\eql@tagwidth@
1667         \advance\dimen@\eql@tagmargin@
1668     \fi
1669     \ifdim\eql@fieldwidth@<\dimen@
1670         \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1671         \eql@tagbox@print@tagsright
1672     \else
1673         \eql@adjust@try\eql@tagwidth@
1674         \ifnum\badness<\eql@maxbadness@
1675             \ifdim\eql@tagwidth@<\eql@tagmargin@
1676                 \eql@adjust@print@alignleft\z@\eql@tagwidth@
1677             \else
1678                 \eql@adjust@print@alignright\z@\eql@tagwidth@
1679             \fi
1680         \eql@tagbox@print@tagsright
1681     \else
1682         \eql@adjust@layoutcenter@tagsright@aligncenter@notag
1683         \eql@tagbox@print@tagsright@raise
1684     \fi
1685 \fi
1686 }

```

TODO: describe

```

1687 \def\eql@adjust@layoutcenter@tagsleft@aligncenter@notag{%
1688     \dimen@\displaywidth
1689     \advance\dimen@-\eql@tagmargin@
1690     \ifdim\eql@fieldwidth@<\dimen@

```

```

1691 \eql@adjust@print@aligncenter\eql@tagmargin@z@
1692 \else
1693 \eql@adjust@print@alignright\z@\z@
1694 \fi
1695 }

```

TODO: describe

```

1696 \def\eql@adjust@layoutcenter@tagsleft@aligncenter@tag{%
1697 \dimen@displaywidth
1698 \ifdim\eql@tagwidth@<\eql@tagmargin@
1699 \advance\dimen@-\eql@tagmargin@
1700 \else
1701 \advance\dimen@-2\eql@tagwidth@
1702 \advance\dimen@\eql@tagmargin@
1703 \fi
1704 \ifdim\eql@fieldwidth@<\dimen@
1705 \eql@tagbox@print@tagsleft
1706 \eql@adjust@print@aligncenter\eql@tagmargin@z@
1707 \else
1708 \eql@adjust@try\eql@tagwidth@
1709 \ifnum\badness<\eql@maxbadness@
1710 \eql@tagbox@print@tagsleft
1711 \ifdim\eql@tagwidth@<\eql@tagmargin@
1712 \eql@adjust@print@alignright\eql@tagwidth@z@
1713 \else
1714 \eql@adjust@print@alignleft\eql@tagwidth@z@
1715 \fi
1716 \else
1717 \eql@tagbox@print@tagsleft@raise
1718 \eql@adjust@layoutcenter@tagsleft@aligncenter@notag
1719 \fi
1720 \fi
1721 \eql@display@firstavail@set\z@
1722 }

```

TODO: describe

```

1723 \def\eql@adjust@layoutcenter@tagsright@alignleft@notag{%
1724 \dimen@displaywidth
1725 \advance\dimen@-\eql@marginleft@
1726 \advance\dimen@-\eql@shape@amount@
1727 \ifdim\eql@fieldwidth@<\dimen@
1728 \dimen@\eql@marginleft@
1729 \advance\dimen@\eql@shape@amount@
1730 \eql@adjust@print@alignleft\dimen@z@
1731 \else
1732 \eql@adjust@print@alignright\z@\z@
1733 \fi
1734 }

```

TODO: describe

```

1735 \def\eql@adjust@layoutcenter@tagsright@alignleft@tag{%
1736 \dimen@\eql@marginleft@
1737 \advance\dimen@\eql@shape@amount@
1738 \advance\dimen@\eql@tagwidth@
1739 \eql@adjust@try\dimen@
1740 \ifnum\badness<\eql@marginbadness@
1741 \dimen@\eql@marginleft@
1742 \advance\dimen@\eql@shape@amount@

```

```

1743 \eql@adjust@print@alignleft\dimen@\eql@tagwidth@
1744 \eql@tagbox@print@tagsright
1745 \else
1746 \ifdim\eql@marginleft@>-\eql@shape@amount@
1747 \eql@adjust@try\eql@tagwidth@
1748 \fi
1749 \ifnum\badness<\eql@maxbadness@
1750 \eql@adjust@print@alignright\z@\eql@tagwidth@
1751 \eql@tagbox@print@tagsright
1752 \else
1753 \eql@adjust@layoutcenter@tagsright@alignleft@notag
1754 \eql@tagbox@print@tagsright@raise
1755 \fi
1756 \fi
1757 }

```

TODO: describe

```

1758 \def\eql@adjust@layoutcenter@tagsleft@alignright@notag{%
1759 \dimen@\displaywidth
1760 \advance\dimen@-\eql@tagmargin@
1761 \advance\dimen@-\eql@marginright@
1762 \ifdim\eql@fieldwidth@<\dimen@
1763 \eql@adjust@print@alignright\z@\eql@marginright@
1764 \else
1765 \eql@adjust@print@alignleft\z@\z@
1766 \fi
1767 }

```

TODO: describe

```

1768 \def\eql@adjust@layoutcenter@tagsleft@alignright@tag{%
1769 \dimen@\eql@marginright@
1770 \advance\dimen@\eql@tagwidth@
1771 \eql@adjust@try\dimen@
1772 \ifnum\badness<\eql@marginbadness@
1773 \eql@tagbox@print@tagsleft
1774 \eql@adjust@print@alignright\eql@tagwidth@\eql@marginright@
1775 \else
1776 \ifdim\eql@marginright@>\z@
1777 \eql@adjust@try\eql@tagwidth@
1778 \fi
1779 \ifnum\badness<\eql@maxbadness@
1780 \eql@tagbox@print@tagsleft
1781 \eql@adjust@print@alignleft\eql@tagwidth@\z@
1782 \else
1783 \eql@tagbox@print@tagsleft@raise
1784 \eql@adjust@layoutcenter@tagsleft@alignright@notag
1785 \fi
1786 \fi
1787 \eql@display@firstavail@set\z@
1788 }

```

TODO: describe

```

1789 \def\eql@adjust@layoutcenter@tagsright@alignright@notag{%
1790 \dimen@\displaywidth
1791 \advance\dimen@-\eql@tagmargin@
1792 \advance\dimen@-\eql@marginright@
1793 \ifdim\eql@fieldwidth@<\dimen@
1794 \dimen@\eql@tagmargin@

```

```

1795 \advance\dimen@\eql@marginright@
1796 \eql@adjust@print@alignright\z@\dimen@
1797 \else
1798 \eql@adjust@print@alignleft\z@\z@
1799 \fi
1800 }

```

TODO: describe

```

1801 \def\eql@adjust@layoutcenter@tagsright@alignright@tag{%
1802 \dimen@\eql@tagmargin@
1803 \advance\dimen@\eql@marginright@
1804 \ifdim\eql@tagwidth@<\dimen@
1805 \eql@adjust@try\dimen@%
1806 \ifnum\badness<\eql@marginbadness@
1807 \eql@adjust@print@alignright\z@\dimen@
1808 \eql@tagbox@print@tagsright
1809 \else
1810 \eql@adjust@try\eql@tagwidth@
1811 \ifnum\badness<\eql@maxbadness@
1812 \eql@adjust@print@alignleft\z@\eql@tagwidth@
1813 \eql@tagbox@print@tagsright
1814 \else
1815 \eql@adjust@print@alignleft\z@\z@
1816 \eql@tagbox@print@tagsleft@raise
1817 \fi
1818 \fi
1819 \else
1820 \eql@adjust@try\eql@tagwidth@
1821 \ifnum\badness<\eql@maxbadness@
1822 \eql@adjust@print@alignright\z@\eql@tagwidth@
1823 \eql@tagbox@print@tagsright
1824 \else
1825 \eql@adjust@layoutcenter@tagsright@alignright@notag
1826 \eql@tagbox@print@tagsright@raise
1827 \fi
1828 \fi
1829 }

```

TODO: describe

```

1830 \def\eql@adjust@layoutcenter@tagsleft@alignleft@notag{%
1831 \dimen@\displaywidth
1832 \advance\dimen@-\eql@tagmargin@
1833 \advance\dimen@-\eql@marginleft@
1834 \advance\dimen@-\eql@shape@amount@
1835 \ifdim\eql@fieldwidth@<\dimen@
1836 \dimen@\eql@tagmargin@
1837 \advance\dimen@\eql@marginleft@
1838 \advance\dimen@\eql@shape@amount@
1839 \eql@adjust@print@alignleft\dimen@\z@
1840 \else
1841 \eql@adjust@print@alignright\z@\z@
1842 \fi
1843 }

```

TODO: describe

```

1844 \def\eql@adjust@layoutcenter@tagsleft@alignleft@tag{%
1845 \dimen@\eql@tagmargin@
1846 \advance\dimen@\eql@marginleft@

```

```

1847 \advance\dimen@ \eql@shape@amount@
1848 \ifdim \eql@tagwidth@ < \dimen@
1849   \eql@adjust@try\dimen@%
1850   \ifnum \badness < \eql@marginbadness@
1851     \eql@tagbox@print@tagsleft
1852     \eql@adjust@print@alignleft\dimen@\z@
1853   \else
1854     \eql@adjust@try\eql@tagwidth@
1855     \ifnum \badness < \eql@maxbadness@
1856       \eql@tagbox@print@tagsleft
1857       \eql@adjust@print@alignright\eql@tagwidth@\z@
1858     \else
1859       \eql@tagbox@print@tagsleft@raise
1860       \eql@adjust@print@alignright\z@\z@
1861     \fi
1862   \fi
1863 \else
1864   \eql@adjust@try\eql@tagwidth@
1865   \ifnum \badness < \eql@maxbadness@
1866     \eql@tagbox@print@tagsleft
1867     \eql@adjust@print@alignleft\eql@tagwidth@\z@
1868   \else
1869     \eql@tagbox@print@tagsleft@raise
1870     \eql@adjust@layoutcenter@tagsleft@alignleft@notag
1871   \fi
1872 \fi
1873 \eql@display@firstavail@set\z@
1874 }

```

eql@adjust@tagmargin

```

1875 \def \eql@adjust@tagmargin{%
1876   \ifdefined \eql@tagmargin@val
1877     \eql@tagmargin@\glueexpr \eql@tagmargin@val \relax
1878   \else
1879     \eql@tagmargin@\eql@tagwidth@max@
1880     \ifdim \eql@tagmargin@ > \z@
1881       \advance \eql@tagmargin@ - \eql@tagsepmin@
1882     \fi
1883   \fi

1884   \dimen@ \eql@tagrows@ \p@
1885   \ifnum \eql@totalrows@ = \@ne
1886     \ifnum \eql@tagrows@ = \@ne
1887       \advance \dimen@ 1sp \relax
1888     \fi
1889   \fi
1890   \ifdim \dimen@ > \eql@totalrows@ \eql@tagmargin@ratio@ \else
1891     \eql@tagmargin@\z@
1892   \fi

1893   \@tempdima \displaywidth
1894   \advance \@tempdima - \eql@totalwidth@
1895   \advance \@tempdima - \eql@columns@inter@ \eql@colsepmin@
1896   \@tempdimb \@tempdima
1897   \advance \@tempdimb - \tw@ \eql@tagmargin@
1898   \ifdim \@tempdimb > \z@
1899     \ifdim \eql@tagmargin@threshold@ \tempdima < \@tempdimb
1900       \eql@tagmargin@\z@

```

```

1901   \fi
1902   \fi
1903 }

```

I.5 Left Alignment Layout

TODO: describe

```

1904 \def\eq@adjust@layoutleft@alignleft{%
1905   \eq@marginleft@eq@layoutleftmargin@
1906   \advance\eq@marginleft@eq@shape@amount@
1907   \ifdim\eq@marginleft@<\eq@layoutleftmarginmin@
1908     \eq@marginleft@eq@layoutleftmarginmin@
1909   \fi
1910   \ifdim\eq@marginleft@>\eq@layoutleftmarginmax@
1911     \eq@marginleft@eq@layoutleftmarginmax@
1912   \fi
1913 }

```

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)

```

1914 \def\eq@adjust@layoutleft@alignleft@notag{%
1915   \ifdim\eq@layoutleftmarginmin@<\eq@marginleft@
1916     \eq@adjust@try\eq@marginleft@
1917     \ifnum\badness<\eq@marginbadness@
1918       \eq@adjust@print@alignleft\eq@marginleft@z@
1919     \else
1920       \eq@adjust@print@alignleft\eq@layoutleftmarginmin@z@
1921     \fi
1922   \else
1923     \eq@adjust@print@alignleft\eq@marginleft@z@
1924   \fi
1925 }

```

TODO: describe

```

1926 \def\eq@adjust@layoutleft@tagsright@alignleft@notag{%
1927   \eq@adjust@layoutleft@alignleft
1928   \eq@adjust@layoutleft@alignleft@notag
1929 }
1930 \let\eq@adjust@layoutleft@tagsleft@alignleft@notag
1931   \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)

```

1932 \def\eq@adjust@layoutleft@tagsright@alignleft@tag{%
1933   \eq@adjust@layoutleft@alignleft
1934   \dimen@eq@marginleft@
1935   \advance\dimen@eq@tagwidth@
1936   \eq@adjust@try\dimen@
1937   \ifnum\badness<\eq@marginbadness@
1938     \eq@adjust@print@alignleft\eq@marginleft@\eq@tagwidth@
1939     \eq@tagbox@print@tagsright
1940   \else
1941     \ifdim\eq@layoutleftmarginmin@<\eq@marginleft@

```

```

1942     \dimen@eql@layoutleftmarginmin@
1943     \advance\dimen@eql@tagwidth@
1944     \eql@adjust@try\dimen@
1945     \fi
1946     \ifnum\badness<\eql@maxbadness@
1947     \eql@adjust@print@alignleft\eql@layoutleftmarginmin@eql@tagwidth@
1948     \eql@tagbox@print@tagsright
1949     \else
1950     \eql@adjust@layoutleft@alignleft@notag
1951     \eql@tagbox@print@tagsright@raise
1952     \fi
1953 \fi
1954 }

1955 \def\eql@adjust@layoutleft@tagsleft@alignleft@tagf%
1956 \eql@adjust@layoutleft@alignleft
1957 \ifdim\eql@tagwidth@<\eql@layoutleftmarginmin@
1958 \eql@tagbox@print@tagsleft
1959 \eql@adjust@layoutleft@notag
1960 \else
1961 \ifdim\eql@tagwidth@<\eql@marginleft@
1962 \eql@adjust@try\eql@marginleft@
1963 \ifnum\badness<\eql@marginbadness@
1964 \eql@tagbox@print@tagsleft
1965 \eql@adjust@print@alignleft\eql@marginleft@z@
1966 \else
1967 \eql@adjust@try\eql@tagwidth@
1968 \ifnum\badness<\eql@maxbadness@
1969 \eql@tagbox@print@tagsleft
1970 \eql@adjust@print@alignleft\eql@tagwidth@z@
1971 \else
1972 \eql@tagbox@print@tagsleft@raise
1973 \eql@adjust@print@alignleft\eql@layoutleftmarginmin@z@
1974 \fi
1975 \fi
1976 \else
1977 \ifdim\eql@tagwidth@>\eql@layoutleftmarginmax@
1978 \eql@tagbox@print@tagsleft@raise
1979 \eql@adjust@layoutleft@alignleft@notag
1980 \else
1981 \eql@adjust@try\eql@tagwidth@
1982 \ifnum\badness<\eql@maxbadness@
1983 \eql@tagbox@print@tagsleft
1984 \eql@adjust@print@alignleft\eql@tagwidth@z@
1985 \else
1986 \eql@tagbox@print@tagsleft@raise
1987 \eql@adjust@layoutleft@alignleft@notag
1988 \fi
1989 \fi
1990 \fi
1991 \fi
1992 \eql@display@firstavail@setz@
1993 }

```

TODO: describe

```

1994 \def\eql@adjust@layoutleft@alignright@notagf%
1995 \eql@marginleft@\eql@layoutleftmargin@
1996 \ifdim\eql@layoutleftmarginmin@<\eql@marginleft@

```

```

1997 \eql@adjust@try\eql@marginleft@
1998 \ifnum\badness<\eql@marginbadness@
1999 \eql@adjust@print@alignright\eql@marginleft@z@
2000 \else
2001 \eql@adjust@print@alignright\eql@layoutleftmarginmin@z@
2002 \fi
2003 \else
2004 \eql@adjust@print@alignright\eql@marginleft@z@
2005 \fi
2006 }
2007 \let\eql@adjust@layoutleft@tagsright@alignright@notag
2008 \eql@adjust@layoutleft@alignright@notag
2009 \let\eql@adjust@layoutleft@tagsleft@alignright@notag
2010 \eql@adjust@layoutleft@alignright@notag

```

TODO: describe

```

2011 \def\eql@adjust@layoutleft@tagsright@alignright@tag{%
2012 \dimen@ \eql@marginleft@
2013 \advance\dimen@ \eql@tagwidth@
2014 \eql@adjust@try\dimen@
2015 \ifnum\badness<\eql@marginbadness@
2016 \eql@adjust@print@alignright\eql@marginleft@ \eql@tagwidth@
2017 \eql@tagbox@print@tagsright
2018 \else
2019 \ifdim\eql@layoutleftmarginmin@<\eql@marginleft@
2020 \dimen@ \eql@layoutleftmarginmin@
2021 \advance\dimen@ \eql@tagwidth@
2022 \eql@adjust@try\dimen@
2023 \fi
2024 \ifnum\badness<\eql@maxbadness@
2025 \eql@adjust@print@alignright\eql@layoutleftmarginmin@ \eql@tagwidth@
2026 \eql@tagbox@print@tagsright
2027 \else
2028 \eql@adjust@layoutleft@alignright@notag
2029 \eql@tagbox@print@tagsright@raise
2030 \fi
2031 \fi
2032 }

```

TODO: describe

```

2033 \def\eql@adjust@layoutleft@tagsleft@alignright@tag{%
2034 \ifdim\eql@tagwidth@<\eql@layoutleftmarginmin@
2035 \eql@tagbox@print@tagsleft
2036 \eql@adjust@layoutleft@alignright@notag
2037 \else
2038 \ifdim\eql@tagwidth@<\eql@marginleft@
2039 \eql@adjust@try\eql@marginleft@
2040 \ifnum\badness<\eql@marginbadness@
2041 \eql@tagbox@print@tagsleft
2042 \eql@adjust@print@alignright\eql@marginleft@z@
2043 \else
2044 \eql@adjust@try\eql@tagwidth@
2045 \ifnum\badness<\eql@maxbadness@
2046 \eql@tagbox@print@tagsleft
2047 \eql@adjust@print@alignright\eql@tagwidth@z@
2048 \else
2049 \eql@tagbox@print@tagsleft@raise
2050 \eql@adjust@print@alignright\eql@layoutleftmarginmin@z@

```



```

2051     \fi
2052     \fi
2053     \else
2054     \ifdim\eql@tagwidth@>\eql@layoutleftmarginmax@
2055     \eql@tagbox@print@tagsleft@raise
2056     \eql@adjust@layoutleft@alignright@notag
2057     \else
2058     \eql@adjust@try\eql@tagwidth@
2059     \ifnum\badness<\eql@maxbadness@
2060     \eql@tagbox@print@tagsleft
2061     \eql@adjust@print@alignright\eql@tagwidth@\z@
2062     \else
2063     \eql@tagbox@print@tagsleft@raise
2064     \eql@adjust@layoutleft@alignright@notag
2065     \fi
2066     \fi
2067     \fi
2068     \fi
2069     \eql@display@firstavail@set\z@
2070 }

2071 \def\eql@adjust@layoutleft@aligncenter{%
2072 \eql@error{shove center not implemented for left alignment}%
2073 }
2074 \let\eql@adjust@layoutleft@tagsright@aligncenter@notag
2075 \eql@adjust@layoutleft@aligncenter
2076 \let\eql@adjust@layoutleft@tagsright@aligncenter@tag
2077 \eql@adjust@layoutleft@aligncenter
2078 \let\eql@adjust@layoutleft@tagsleft@aligncenter@notag
2079 \eql@adjust@layoutleft@aligncenter
2080 \let\eql@adjust@layoutleft@tagsleft@aligncenter@tag
2081 \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
2082 \let\eql@fieldlength@tab@empty
2083 \def\eql@fieldlength@save#1{%
2084 \begingroup
2085 \let\or\relax
2086 \global\edef\eql@fieldlength@tab{%
2087 \eql@fieldlength@tab
2088 \ifnum#1=\@ne
2089 \or
2090 \else
2091 ,%
2092 \fi
2093 \the\wd\eql@fieldbox@

```

```

2094   }%
2095  \endgroup
2096 }
2097 \def\eql@fieldlength@get#1{%
2098  \ifcase\expandafter#1\eql@fieldlength@tab\fi
2099 }

```

```

\eql@tagwidth@get
\eql@tagwidth@save

```

```

2100 \let\eql@tagwidth@tab\@empty
2101 \def\eql@tagwidth@get#1{%
2102  \ifcase\expandafter#1\eql@tagwidth@tab\fi
2103 }
2104 \def\eql@tagwidth@save{%
2105  \begingroup
2106   \let\or\relax
2107   \global\edef\eql@tagwidth@tab{\eql@tagwidth@tab\or\the\eql@tagwidth@}%
2108  \endgroup
2109 }
2110 \def\eql@tagwidth@savezero{%
2111  \begingroup
2112   \let\or\relax
2113   \global\edef\eql@tagwidth@tab{\the\eql@tagwidth@\eql@tagwidth@tab}%
2114  \endgroup
2115 }

```

```

\eql@colwidth@tab

```

```

2116 \let\eql@colwidth@tab\@empty

```

```

columns@colwidth@get

```

```

2117 \def\eql@columns@colwidth@get#1{%
2118  \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2119 }
2120 \def\eql@columns@colwidth@save{%
2121  \begingroup
2122   \let\or\relax
2123   \global\edef\eql@colwidth@tab{\or\the\wd\thr@\eql@colwidth@tab}%
2124  \endgroup
2125 }

```

J.2 Measure Support

TODO: describe

```

2126 \def\eql@measure@init#1{%
2127  \measuring@true
2128  \eql@row@\z@
2129  \let\displaybreak\eql@displaybreak@measure
2130  \tabskip\z@skip
2131  \everycr{%
2132   \noalign{%
2133    \global\advance\eql@row@\@ne
2134    #1%
2135   }%
2136  }%
2137 }

```

```

sure@restorecounters
measure@savecounters
2138 \let\eqL@measure@restorecounters\@empty
2139 \def\eqL@measure@savecounters{%
2140   \begingroup
2141     \def\@elt##1{%
2142       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2143     \global\edef\@gtempa{%
2144       \cl@ckpt
2145       \let\noexpand\eqL@measure@restorecounters\noexpand\@empty
2146     }%
2147   \endgroup
2148   \let\eqL@measure@restorecounters\@gtempa
2149 }

```

J.3 Print Support

TODO: describe

```

eqL@print@inithalign
2150 \def\eqL@print@init#1{%
2151   \eqL@row@z@
2152   \eqL@halign@init{%
2153     \global\eqL@displaybreak@open@\@MM
2154     \global\advance\eqL@row@\@ne
2155     #1%
2156   }%
2157 }

2158 \def\eqL@print@overfull{%
2159   \dimen@\eqL@line@width@
2160   \advance\dimen@-\hfuzz
2161   \ifdim\dimen@>\displaywidth
2162     \setbox\z@\hbox to\displaywidth{\hbox to\eqL@line@width@{\hfil}}%
2163     \wd\z@\z@
2164     \ht\z@\eqL@line@height@
2165     \dp\z@\eqL@line@depth@
2166     \box\z@
2167   \fi
2168 }

```

```

l@tagbox@print@multi
2169 \def\eqL@tagbox@print@multi{%
2170   \advance\eqL@tagwidth@-\eqL@tagfuzz@
2171   \ifdefined\eqL@tagsleft
2172     \eqL@display@firstavail@set\z@
2173     \ifdim\eqL@tagwidth@>\eqL@line@avail@
2174       \eqL@tagbox@print@tagsleft@raise
2175     \else
2176       \eqL@tagbox@print@tagsleft
2177     \fi
2178     \kern\displaywidth
2179   \else
2180     \kern\displaywidth
2181     \advance\eqL@tagwidth@\eqL@line@width@
2182     \ifdim\eqL@tagwidth@>\displaywidth

```

```

2183     \eql@tagbox@print@tagsright@raise
2184     \else
2185     \eql@tagbox@print@tagsright
2186     \fi
2187     \fi
2188 }

```

J.4 Line Breaks

TODO: describe

`\eql@math@cr`

```

2189 \protected\def\eql@math@cr{%
2190     \eql@ampprotecttwo\eql@teststaropt@tight
2191     {\global\eql@displaybreak@pen@ \@M\eql@math@cr}\eql@math@cr@z@}

```

`\eql@math@cr@`

```

2192 \def\eql@math@cr@[#1]{%
2193     \eql@math@cr@@@
2194     \cr
2195     \noalign{%
2196         \ifnum\eql@displaybreak@pen@=\@MM
2197             \penalty\interdisplaylinepenalty
2198         \else
2199             \penalty\eql@displaybreak@pen@
2200         \fi
2201         \advance\eql@vspaceskip@\glueexpr#1\relax%
2202         \vskip\eql@vspaceskip@
2203         \global\eql@vspaceskip@\z@skip
2204     }%
2205 }

```

`\eql@let@cr`

```

2206 \def\eql@let@cr#1{%
2207     \let\\eql@math@cr
2208     \let\eql@math@cr@@@#1%
2209 }

```

J.5 Intertext

TODO: describe

TODO: revert in everymath?

```

2210 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2211 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2212 \def\eql@intertext@process{%
2213     \eql@math@cr@@@
2214     \cr
2215     \ifmeasuring@
2216         \expandafter\@gobble

```

```

2217 \else
2218   \expandafter\eql@intertext@print
2219 \fi
2220 }

```

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?

```

2221 \def\eql@intertext@print#1{%
2222   \noalign{%
2223     \eql@halign@after
2224     \let\eql@skip@force@below\z@
2225     \let\eql@skip@force@above\z@
2226     \eql@setkeys{intertext}\eql@intertext@opt
2227     \openup-\eql@spread@
2228     \penalty\postdisplaypenalty
2229     \ifcase\eql@skip@force@below\relax
2230       \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2231     \or
2232       \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2233     \or
2234       \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2235     \or
2236       \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2237     \or
2238       \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2239     \or
2240       \advance\eql@vspaceskip@\z@skip
2241     \or
2242       \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2243     \or
2244       \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2245     \fi
2246     \vskip\eql@vspaceskip@
2247     \global\eql@vspaceskip@\z@skip
2248     \vbox{%
2249       \@parboxrestore
2250       \ifdim
2251         \ifdim@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2252       \else
2253         \parshape@one
2254         \@totalleftmargin\linewidth
2255       \fi
2256       \noindent
2257       \ignorespaces
2258       #1%
2259       \par
2260     }%
2261     \penalty\predisplaypenalty
2262     \ifcase\eql@skip@force@above\relax
2263       \vskip\glueexpr\eql@skip@long@above\relax
2264     \or
2265       \vskip\glueexpr\eql@skip@short@above\relax
2266     \or
2267       \vskip\glueexpr\eql@skip@cont@above\relax
2268     \or
2269       \vskip\glueexpr\eql@skip@par@above\relax
2270     \or
2271       \vskip\glueexpr\eql@skip@top@above\relax

```

```

2272 \or
2273 \vskip\z@skip
2274 \or
2275 \vskip\glueexpr\eq@skip@med@above\relax
2276 \or
2277 \vskip\glueexpr\eq@skip@custom@above\relax
2278 \fi
2279 % \eq@halign@prevdepth@maxdimen
2280 \eq@halign@prevdepth@z@
2281 \eq@halign@before
2282 }
2283 }

```

TODO: describe

```

2284 \newenvironment{eq@intertext}{%
2285 \eq@testopt@tight\eq@intertext@{}}%
2286 }{%
2287 \aftergroup\eq@intertext@after
2288 \ignorespacesafterend
2289 }

```

TODO: describe

```

2290 \def\eq@intertext@env{intertext}
2291 \def\eq@intertext@[#1]{%
2292 \global\def\eq@intertext@opt{#1}%
2293 \ifx\@currentvir\eq@intertext@env
2294 \expandafter\eq@scan@env\expandafter\eq@intertext@inject
2295 \else
2296 \expandafter\eq@intertext@process
2297 \fi
2298 }

```

TODO: describe

```

2299 \def\eq@intertext@inject{%
2300 \global\edef\eq@intertext@after{%
2301 \noexpand\eq@intertext@process{%
2302 \ifx\eq@scan@body\eq@scan@body@dump
2303 \eq@scan@body@dump
2304 \else
2305 \noexpand\scantokens{\eq@scan@body@dump}%
2306 \fi
2307 }%
2308 }%
2309 }

```

K Equations Box Environment

TODO: describe

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

K.1 Line Breaks

`\eq@box@cr`

```

2310 \protected\def\eqL@box@cr{%
2311   \eqL@amprotect\eqL@testopt@tight\eqL@box@cr@\z@
2312 }
2313 \def\eqL@box@cr@[#1]{%
2314   \eqL@punct@apply@line
2315   \eqL@hook@lineout
2316   \eqL@box@lastfield
2317   \cr
2318   \noalign{%
2319     \vskip\glueexpr#1\relax
2320   }%
2321 }

```

K.2 Stacked Mode

```

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

```

```

2365     \box\eql@fieldbox@
2366     \skip@\@flushglue
2367     \advance\skip@\eql@paddingright@\relax
2368     \hskip\skip@
2369     \eql@tagging@aligncenter
2370     \or
2371     \skip@\@flushglue
2372     \advance\skip@\eql@paddingleft@\relax
2373     \advance\skip@\eql@paddingright@\relax
2374     \hskip\skip@
2375     \box\eql@fieldbox@
2376     \eql@tagging@alignright
2377     \fi
2378     \tabskip\eql@colsep@\relax
2379     \crrc
2380     \noalign{%
2381       \eql@hook@blockbefore
2382     }%
2383     \eql@hook@blockin
2384 }
2385 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

K.3 Aligned Mode

```

2386 \def\eql@box@lastfield@odd{%
2387   &\omit
2388   \kern-\wd\eql@fieldbox@
2389   \box\eql@fieldbox@
2390   \hfil
2391   &\omit\kern-\eql@colsep@
2392 }%
2393 \def\eql@box@lastfield@even{&\omit\kern-\eql@colsep@}

2394 \def\eql@box@open@aligned{%
2395   % \TODO templates
2396   \let\eql@box@lastfield\@empty
2397   \eql@halign@init{%
2398     (dev)\eql@dev{starting new line}%
2399   }%
2400   \tabskip\z@skip
2401   \halign\bgroup
2402     &%
2403     \let\eql@box@lastfield\eql@box@lastfield@odd
2404     \global\setbox\eql@fieldbox@\hbox{%
2405       \eql@strut@field
2406       \@lign
2407       $\m@th\displaystyle
2408       \eql@hook@colin
2409       ##%
2410       \eql@class@innerleft
2411       \eql@hook@innerleft
2412       \eql@tagging@mathsave
2413       $%
2414       \eql@tagging@mathaddlast
2415     }%
2416     \hfil
2417     \kern\wd\eql@fieldbox@
2418     \tabskip\z@skip
2419     &%

```



```

2420 \eqldfieldwidth@wd\eqldfieldbox@
2421 \kern-\eqldfieldwidth@
2422 \box\eqldfieldbox@
2423 \let\eqldbox@lastfield\eqldbox@lastfield@even
2424 \llap{\unhbox\eqldfieldbox@}%
2425 \hbox{%
2426 \eqldstrut@field
2427 \@lign
2428 $\m@th\displaystyle
2429 \eqldhook@innerright
2430 \eqldclass@innerright@sel
2431 ##%
2432 \eqldpunct@apply@col
2433 \eqldhook@colout
2434 \eqldtagging@mathsave
2435 $%
2436 \eqldtagging@mathaddlast
2437 }%
2438 \hfil
2439 \tabskip\eqldcolsep@\relax
2440 \crrr
2441 \noalign{%
2442 \eqldhook@blockbefore
2443 }%
2444 \eqldhook@blockin
2445 }
2446 \def\eqldmode@aligned{\let\eqldbox@open\eqldbox@open@aligned}

```

K.4 Main

```

2447 \let\eqldbox@box\vcenter
2448 \let\eqldbox@open\@undefined
2449 \def\eqldbox@close{%
2450 \ifvmode\else
2451 \global\eqldtotalrows@\eqldrow@
2452 \eqldpunct@apply@block
2453 \eqldbox@cr@[\z@skip]%
2454 \fi
2455 \crrr
2456 \noalign{%
2457 \eqldhook@blockafter
2458 }%
2459 \eqldtagging@tablesaveinner
2460 \egroup
2461 }

```

\eqldbox@start

```

2462 \def\eqldbox@start{%
2463 \relax
2464 \ifmmode
2465 \let\eqldbox@endmath\@empty
2466 \else
2467 $\let\eqldbox@endmath=$%
2468 \fi
2469 \eqldnextopt@process{equationsbox}%
2470 \let\eqldpunct@block\eqldpunct@main
2471 \let\eqldpunct@main\relax
2472 \eqldcolsep@\glueexpr\eqldbox@colsep\relax

```

```

2473 \eql@paddingleft@glueexpr\eql@paddingleft@val\relax
2474 \eql@paddingright@glueexpr\eql@paddingright@val\relax
2475 \eql@indent@glueexpr\eql@indent@val\relax
2476 \eql@stack@save@boxed
2477 \let\eql@layoutleft\eql@false
2478 \eql@row@z@
2479 \eql@totalrows@M
2480 \eql@shape@sel
2481 \hskip\glueexpr\eql@box@marginleft\relax
2482 \eql@box@box\bgroup
2483 \eql@display@leave
2484 \let\\eql@box@cr
2485 \eql@box@open
2486 }

```

`\eql@box@end`

```

2487 \newcommand{\eql@box@end}{%
2488 \eql@box@close
2489 \egroup
2490 \eql@tagging@tableaddinner
2491 \hskip\glueexpr\eql@box@marginright\relax
2492 \eql@stack@restore
2493 \eql@box@endmath
2494 }

```

K.5 Environment

`equationsbox` (*env.*)

```

2495 \newenvironment{equationsbox}{%
2496 (dev)\eql@dev@enterenv
2497 \eql@ampprotect\eql@box@testall\eql@box@start
2498 }{%
2499 \eql@box@end
2500 (dev)\eql@dev@leaveenv
2501 }

2502 \def\eql@box@testall{\eql@box@testtilde}
2503 \def\eql@box@testtilde#1{%
2504 \eql@ifnextgobble@tight~%
2505 {\eqnadopt{lines}\eql@box@testopt{#1}}%
2506 {\eql@box@testopt{#1}}}
2507 \def\eql@box@testopt#1{%
2508 \eql@ifnextchar@tight[%]
2509 {\eql@box@adopt{#1}}%
2510 {#1}}
2511 \def\eql@box@adopt#1[#2]{\eqnadopt{#2}#1}

```

L Single-Line Equation

TODO: describe

L.1 Native Mode

```

2512 \def\eql@single@start@native{%

```

```

2513 % \mathopen{}%
2514 \eql@hook@eqin
2515 }%

```

TODO: describe

```

2516 \def\eql@single@end@native{%
2517 % \mathclose{}%
2518 \if@eqnsw
2519 \ifdefined\eql@tagsleft
2520 \leqno
2521 \else
2522 \eqno
2523 \fi
2524 \eql@compose@print
2525 \fi
2526 \ifnum\eql@displaybreak@open@=\@MM\else
2527 \postdisplaypenalty\eql@displaybreak@open@
2528 \fi
2529 }%

```

L.2 Print

```

2530 \def\eql@single@start@print{%
2531 \eql@totalrows@\@ne
2532 \eql@row@\z@
2533 \eql@display@init
2534 \let\shoveleft\eql@adjust@alignleft
2535 \let\shovecenter\eql@adjust@aligncenter
2536 \let\shoveright\eql@adjust@alignright
2537 \eql@adjust@init
2538 \eql@shape@eval
2539 \eql@halign@init{}%

2540 \eql@row@\@ne
2541 \setbox\eql@fieldbox@\hbox\bgroup
2542 \eql@restore@hfuzz
2543 \eql@strut@field
2544 $\m@th\displaystyle%$
2545 \eql@hook@eqin
2546 }

2547 \def\eql@single@end@print{%
2548 \eql@tagging@mathsave
2549 $$
2550 \hfil
2551 \kern\z@
2552 \egroup

2553 \eql@fieldwidth@\wd\eql@fieldbox@
2554 \eql@line@height@\ht\eql@fieldbox@
2555 \eql@line@depth@\dp\eql@fieldbox@
2556 \eql@totalwidth@\eql@fieldwidth@
2557 \eql@totalrows@\@ne

2558 \if@eqnsw
2559 \eql@tagbox@make\eql@compose@print
2560 \eql@tagrows@\@ne
2561 \else
2562 \eql@tagwidth@\z@
2563 \eql@tagrows@\z@
2564 \fi

```

```

2565 \eql@tagwidth@max@\eql@tagwidth@
2566 \eql@adjust@calc
2567 \halign{##\cr
2568 \noalign{\eql@halign@before}%
2569 \if@eqnsw
2570 \eql@adjust@sel@tag
2571 \else
2572 \eql@adjust@sel@notag
2573 \fi
2574 \cr
2575 \noalign{\eql@halign@after}%
2576 \eql@tagging@tablesavelines
2577 }%
2578 \eql@row@tw@
2579 \eql@display@close
2580 }

```

M Multi-Line Lines Mode

M.1 Measure

TODO: describe

```

2581 \def\eql@lines@measure@line@begin{%
2582 (dev)\eql@dev{starting line \the\eql@row@}%
2583 \eql@numbering@measure@line@begin
2584 \eql@hook@linein
2585 }

```

TODO: describe

```

2586 \def\eql@lines@measure@line@end{%
2587 \eql@punct@apply@line
2588 \eql@hook@lineout
2589 }

```

TODO: describe

```

2590 \def\eql@lines@measure@field{%
2591 \kern\wd\eql@fieldbox@
2592 }

```

TODO: describe

```

2593 \def\eql@lines@measure@tag{%
2594 \ifnum\eql@numbering@target@<\z@
2595 \if@eqnsw
2596 \eql@tagbox@make\eql@compose@measure
2597 \ifdim\eql@tagwidth@>\eql@tagwidth@max@
2598 \global\eql@tagwidth@max@\eql@tagwidth@
2599 \fi
2600 \global\advance\eql@tagrows@\@ne
2601 \else
2602 \eql@tagwidth@\z@
2603 \fi
2604 \fi
2605 }

```

`\eql@lines@measure`

```
2606 \def\eql@lines@measure{%
2607 (dev)\eql@dev@center\eql@lines@measure
2608 \eql@tagwidth@max@z@
2609 \eql@tagrows@z@
2610 \eql@measure@savecounters
2611 \setboxz@\vbox{%
2612 \eql@numbering@measure@init
2613 \eql@measure@init\eql@lines@measure@line@begin
2614 \eql@let@cr\eql@lines@measure@line@end
2615 \halign{%
2616 \setbox\eql@fieldbox@\hbox{%
2617 \@lign
2618 $\m@th\displaystyle
2619 \eql@hook@colin
2620 ##%
2621 \eql@punct@apply@col
2622 \eql@hook@colout
2623 $%
2624 }%
2625 \eql@lines@measure@field
2626 \eql@lines@measure@tag
2627 \crr
2628 \noalign{%
2629 \eql@hook@blockbefore
2630 }%
2631 \eql@hook@blockin
2632 \eql@scan@body
2633 \ifvmode\else
2634 \eql@punct@apply@block
2635 \eql@hook@blockout
2636 \eql@lines@measure@line@end
2637 \cr
2638 \fi
2639 \omit
2640 \cr
2641 \noalign{%
2642 \eql@hook@blockafter
2643 }%
2644 }%
2645 \global\advance\eql@row@-\tw@
2646 \eql@numbering@measure@eval
2647 \ifnum\eql@numbering@target@>z@
2648 \eql@tagbox@make\eql@compose@measure
2649 \global\eql@tagwidth@max@\eql@tagwidth@
2650 \global\eql@tagrows@\@ne
2651 \fi
2652 }%
2653 \eql@totalrows@\eql@row@
2654 \ifdefined\eql@numbering@subeq@use
2655 \eql@numbering@subeq@test
2656 \fi
2657 \eql@measure@restorecounters
2658 \setboxz@\vbox{%
2659 \unvboxz@
2660 \unpenalty
2661 \global\setbox@\@ne\lastbox
2662 }%
```

```

2663 \eql@totalwidth@\wd\@ne
2664 (dev)\eql@dev@leave\eql@lines@measure
2665 }

```

M.2 Print

TODO: describe

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

```

2666 \def\eql@lines@print@line@begin{%
2667 (dev)\eql@dev{starting line \the\eql@row}%
2668 \eql@numbering@print@line@begin
2669 \eql@hook@linein
2670 }

```

TODO: describe

```

2671 \def\eql@lines@print@line@end{%
2672 \eql@punct@apply@line
2673 \eql@hook@lineout
2674 }

```

TODO: describe

```

2675 \def\eql@lines@print@line@adjust{%
2676 \eql@numbering@print@line@eval
2677 \eql@fieldwidth@\wd\eql@fieldbox@
2678 \eql@line@height@\ht\eql@fieldbox@
2679 \eql@line@depth@\dp\eql@fieldbox@
2680 \if@eqnsw
2681 \eql@tagbox@make\eql@compose@print
2682 \eql@adjust@sel@tag
2683 \else
2684 \eql@adjust@sel@notag
2685 \fi
2686 }

```

TODO: describe

```

2687 \def\eql@lines@print{%
2688 (dev)\eql@dev@enter\eql@lines@print
2689 \eql@display@init
2690 \eql@adjust@init
2691 \eql@adjust@calc
2692 \eql@numbering@print@init
2693 \eql@print@init\eql@lines@print@line@begin
2694 \eql@let@cr\eql@lines@print@line@end
2695 \tabskip\z@skip
2696 \halign{%
2697 \eql@shape@eval
2698 \setbox\eql@fieldbox@\hbox{%
2699 \eql@restore@hfuzz
2700 \eql@strut@field
2701 \@lign
2702 $\m@th\displaystyle
2703 \eql@hook@colin
2704 ##%
2705 \eql@punct@apply@col

```

```

2706         \eql@hook@colout
2707         \eql@tagging@mathsave
2708         $%
2709         \hfil
2710         \kern\z@
2711     }%
2712     \eql@lines@print@line@adjust
2713     \crr
2714     \noalign{%
2715         \eql@halign@before
2716         \eql@numbering@print@block@begin
2717         \eql@hook@blockbefore
2718     }%
2719 % \TODO relax? leavevmode?!
2720     \eql@hook@blockin
2721     \eql@scan@body
2722     \ifvmode\else
2723         \eql@punct@apply@block
2724         \eql@hook@blockout
2725         \eql@lines@print@line@end
2726         \cr
2727     \fi
2728     \noalign{%
2729         \eql@hook@blockafter
2730         \eql@halign@after
2731 (dev)\eql@dev@leave\eql@lines@print
2732     }%
2733     \eql@tagging@tablesavelines
2734 }%
2735 \eql@display@close
2736 }

```

N Multi-Line Columns Mode

TODO: describe

N.1 Columns Processing

TODO: describe

```

\eql@columns@add@amp
@columns@completerow
2737 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
2738 \def\eql@columns@completerow{%
2739     \count@\eql@totalcolumns@
2740     \advance\count@-\eql@column@
2741     \advance\count@\@ne
2742     \edef\eql@tmpf{%
2743         \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
2744     \eql@tmp
2745 }

```

N.2 Measure

TODO: describe

s@measure@line@begin

```
2746 \def\eql@columns@measure@line@begin{%
2747 (dev)\eql@dev{starting line \the\eql@row}%
2748 \global\eql@column@z@
2749 \eql@numbering@measure@line@begin
2750 \eql@hook@linein
2751 }
```

```
2752 \def\eql@columns@measure@field{%
2753 \eql@fieldlength@save\eql@column@
2754 \kern\wd\eql@fieldbox@
2755 }
```

mns@measure@line@end

```
2756 \def\eql@columns@measure@line@end{%
2757 \eql@punct@apply@line
2758 \eql@hook@lineout
2759 &\omit
2760 \ifnum\eql@column@>\eql@totalcolumns@
2761 \global\eql@totalcolumns@\eql@column@
2762 \fi
2763 \eql@columns@measure@tag
2764 }
```

@columns@measure@tag

```
2765 \def\eql@columns@measure@tag{%
2766 \ifnum\eql@numbering@target@<\z@
2767 \if@eqnsw
2768 \eql@tagbox@make\eql@compose@measure
2769 \ifdim\eql@tagwidth@>\eql@tagwidth@max@
2770 \global\eql@tagwidth@max@\eql@tagwidth@
2771 \fi
2772 \global\advance\eql@tagrows@\@ne
2773 \else
2774 \eql@tagwidth@\z@
2775 \fi
2776 \eql@tagwidth@save
2777 \fi
2778 }
```

\eql@columns@measure

```
2779 \def\eql@columns@measure{%
2780 (dev)\eql@dev@center\eql@columns@measure
2781 \eql@totalcolumns@\z@
2782 \eql@tagwidth@max@\z@
2783 \let\eql@tagwidth@tab\@empty
2784 \let\eql@fieldlength@tab\@empty
2785 \eql@tagrows@\z@
2786 \eql@measure@savecounters
2787 \setbox\z@\vbox{%
2788 \eql@numbering@measure@init
2789 \eql@measure@init\eql@columns@measure@line@begin
2790 \eql@let@cr\eql@columns@measure@line@end
2791 \tabskip\z@skip
2792 \halign{%
```



```

2793 &%
2794 \global\advance\eql@column@\@ne
2795 \hfil
2796 \global\setbox\eql@fieldbox@\hbox{%
2797 \@lign
2798 $\m@th\displaystyle
2799 \eql@hook@colin
2800 ##%
2801 \eql@class@innerleft
2802 \eql@hook@innerleft
2803 $%
2804 }%
2805 \global\eql@fieldwidth@\wd\eql@fieldbox@
2806 \eql@columns@measure@field
2807 &%
2808 \global\advance\eql@column@\@ne
2809 \setbox\eql@fieldbox@\hbox{%
2810 \@lign
2811 $\m@th\displaystyle
2812 \eql@hook@innerright
2813 \eql@class@innerright@sel
2814 ##%
2815 \eql@punct@apply@col
2816 \eql@hook@colout
2817 $%
2818 }%
2819 \eql@columns@measure@field
2820 \hfil
2821 \crr
2822 \noalign{%
2823 \eql@hook@blockbefore
2824 }%
2825 \eql@hook@blockin
2826 \eql@scan@body

```

TODO: test for vmode okay?!

```

2827 \ifvmode\else
2828 \eql@punct@apply@block
2829 \eql@hook@blockout
2830 \eql@columns@measure@line@end
2831 \cr
2832 \fi
2833 \noalign{%
2834 \eql@hook@blockafter
2835 }%

```

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

```

2836 \eql@columns@completerow
2837 \cr
2838 }%
2839 \global\advance\eql@row@-\tw@
2840 \eql@numbering@measure@eval
2841 \ifnum\eql@numbering@target@>\z@
2842 \eql@tagbox@make\eql@compose@measure
2843 \global\eql@tagwidth@max@\eql@tagwidth@
2844 \global\eql@tagrows@\@ne
2845 \eql@tagwidth@savezero

```

```

2846   \fi
2847 }%
2848 \eql@totalrows@\eql@row@
2849 \ifdefined\eql@numbering@subeq@use
2850   \eql@numbering@subeq@test
2851 \fi
2852 \eql@measure@restorecounters
2853 % \eql@totalwidth@\wd\z@

2854 \setbox\z@\vbox{%
2855   \unvbox\z@
2856   \unpenalty
2857   \global\setbox\@ne\lastbox
2858 }%
2859 \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

2860 \let\eql@colwidth@tab\@empty
2861 \loop
2862   \setbox\@ne\hbox{%
2863     \unhbox\@ne
2864     \unskip
2865     \global\setbox\thr@\@lastbox
2866   }%
2867 \ifhbox\thr@@
2868   \eql@columns@colwidth@save
2869 \repeat

2870 (dev)\eql@dev@leave\eql@columns@measure
2871 }

```

N.3 Print

TODO: describe

`\eql@columns@print@line@begin`

```

2872 \def\eql@columns@print@line@begin{%
2873 (dev)\eql@dev{starting line \the\eql@row@}%
2874 \global\eql@column@\z@
2875 \global\eql@line@pos@\eql@marginleft@
2876 \global\eql@line@width@\z@
2877 \global\eql@line@avail@\eql@totalwidth@
2878 \global\eql@line@height@\z@
2879 \global\eql@line@depth@\z@
2880 \eql@numbering@print@line@begin
2881 \eql@hook@linein
2882 }

```

`\eql@columns@print@field`

```

2883 \def\eql@columns@print@field{%

```

determine available and used space

```

2884 \dimen@\eql@columns@colwidth@get\eql@column@\relax
2885 \ifdim\wd\eql@fieldbox@>\z@
2886   \ifdim\eql@line@width@=\z@

```

```

2887 \eq@line@avail@\eq@line@pos@
2888 \ifodd\eq@column@
2889 \advance\eq@line@avail@\dimen@
2890 \advance\eq@line@avail@-\wd\eq@fieldbox@
2891 \fi
2892 \global\eq@line@avail@\eq@line@avail@
2893 \fi
2894 \eq@line@width@\eq@line@pos@
2895 \ifodd\eq@column@
2896 \advance\eq@line@width@\dimen@
2897 \else
2898 \advance\eq@line@width@\wd\eq@fieldbox@
2899 \fi
2900 \global\eq@line@width@\eq@line@width@
2901 \fi
2902 \advance\eq@line@pos@\dimen@
2903 \ifodd\eq@column@\else
2904 \advance\eq@line@pos@\eq@colsep@
2905 \fi
2906 \global\eq@line@pos@\eq@line@pos@

```

update height and depth

```

2907 \ifdim\ht\eq@fieldbox@>\eq@line@height@
2908 \global\eq@line@height@\ht\eq@fieldbox@
2909 \fi
2910 \ifdim\dp\eq@fieldbox@>\eq@line@depth@
2911 \global\eq@line@depth@\dp\eq@fieldbox@
2912 \fi

```

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

```

2913 %
2914 % \box\eq@fieldbox@
2915 %
2916 % \dimen@\eq@columns@colwidth@get\eq@column@\relax
2917 % \advance\dimen@-\wd\eq@fieldbox@
2918 % \ifodd\eq@column@
2919 % \kern\dimen@
2920 % \box\eq@fieldbox@
2921 % \else
2922 % \box\eq@fieldbox@
2923 % \kern\dimen@
2924 % \fi
2925 %
2926 \dimen@\eq@columns@colwidth@get\eq@column@\relax
2927 \ifodd\eq@column@
2928 \kern\dimen@
2929 \else
2930 \advance\dimen@-\wd\eq@fieldbox@
2931 \box\eq@fieldbox@
2932 \kern\dimen@
2933 \fi
2934 %
2935 }

2936 \def\eq@columns@print@trailright{%
2937 &\omit
2938 \global\advance\eq@column@\@ne

```

```

2939 \setbox\eql@fieldbox@\hbox{%
2940 \kern-\wd\eql@fieldbox@\box\eql@fieldbox@
2941 }%
2942 \eql@columns@print@field
2943 }

```

lums@print@line@end

```

2944 \def\eql@columns@print@line@end{%
2945 \eql@punct@apply@line
2946 \eql@hook@lineout
2947 % \TODO add an even column with empty stuff if box processing deferred
2948 \ifodd\eql@column@
2949 \expandafter\eql@columns@print@trailright
2950 \fi
2951 \eql@columns@completerow
2952 \eql@columns@print@tag
2953 }

```

ql@columns@print@tag

```

2954 \def\eql@columns@print@tag{%
2955 \dimen@\eql@totalwidth@
2956 \advance\dimen@\eql@colsep@
2957 \kern-\dimen@

```

determine first line available space

```

2958 \eql@display@firstavail@set\eql@line@avail@
2959 \eql@print@overfull
2960 \eql@numbering@print@line@eval
2961 \if@eqnsw
2962 \eql@tagbox@make\eql@compose@print
2963 \eql@tagging@tagaddbox
2964 \eql@tagbox@print@multi
2965 \else
2966 \eql@tagging@tagaddbox
2967 \kern\displaywidth
2968 \fi
2969 }

```

\eql@columns@print

```

2970 \def\eql@columns@print{%
2971 (dev)\eql@dev@enter\eql@columns@print
2972 \eql@columns@adjust
2973 \eql@display@init
2974 \eql@numbering@print@init
2975 \eql@print@init\eql@columns@print@line@begin
2976 \eql@let@cr\eql@columns@print@line@end
2977 \tabskip\eql@marginleft@
2978 \halign{%
2979 &%
2980 \global\advance\eql@column@\@ne
2981 \hfil
2982 \global\setbox\eql@fieldbox@\hbox{%
2983 \eql@strut@field
2984 \@lign
2985 $\m@th\displaystyle

```

```

2986         \eql@hook@colin
2987         ##%
2988         \eql@class@innerleft
2989         \eql@hook@innerleft
2990         \eql@tagging@mathsave
2991         $%
2992         \eql@tagging@mathaddlast
2993     }%
2994     \global\eql@fieldwidth@wd\eql@fieldbox@
2995     \eql@columns@print@field
2996     \tabskip\z@skip
2997     &%
2998     \global\advance\eql@column@\@ne
2999     \setbox\eql@fieldbox@\hbox{%
3000 % \TODO printing left field in right field
3001     \kern-\wd\eql@fieldbox@
3002     \box\eql@fieldbox@
3003     \eql@strut@field
3004     \@lign
3005     $\m@th\displaystyle
3006     \eql@hook@innerright
3007     \eql@class@innerright@sel
3008     ##%
3009     \eql@punct@apply@col
3010     \eql@hook@colout
3011     \eql@tagging@mathsave
3012     $%
3013     \eql@tagging@mathaddlast
3014     }%
3015     \eql@columns@print@field
3016     \hfil
3017     \tabskip\eql@colsep@\relax
3018     \crrc
3019     \noalign{%
3020     \eql@halign@before
3021     \eql@numbering@print@block@begin
3022     \eql@hook@blockbefore
3023     }%
3024     \eql@hook@blockin
3025     \eql@scan@body
3026     \ifvmode\else
3027     \eql@punct@apply@block
3028     \eql@hook@blockout
3029     \eql@columns@print@line@end
3030     \cr
3031     \fi
3032     \noalign{%
3033     \eql@hook@blockafter
3034     \eql@halign@after
3035 (dev)\eql@dev@leave\eql@columns@print
3036     }%
3037     \eql@tagging@tablesalign
3038     }%
3039     \eql@display@close
3040 }

```

N.4 Adjust

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

```
3041 \def\eql@columns@adjust{%
3042   \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

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

```
3043   \ifodd\eql@totalcolumns@
3044     \advance\eql@totalcolumns@\@ne
3045   \fi
```

TODO: should we guard against no columns?!

```
3046   \ifnum\eql@totalcolumns@<\thr@@
3047     \let\eql@columns@margins\eql@true
3048   \fi
```

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

```
3049   \eql@columns@inter@\eql@totalcolumns@
3050   \divide\eql@columns@inter@\tw@
3051   \advance\eql@columns@inter@\m@ne

3052   \ifdefined\eql@layoutleft
3053     \eql@layoutleftmargin@\glueexpr\eql@layoutleftmargin@val\relax
3054   \else
3055     \eql@adjust@tagmargin
3056   \fi

3057   \eql@colsep@\displaywidth
3058   \advance\eql@colsep@-\eql@totalwidth@
3059   \ifdefined\eql@layoutleft
3060     \advance\eql@colsep@-\eql@layoutleftmargin@
3061   \else
3062     \advance\eql@colsep@-\eql@tagmargin@
3063     \ifdefined\eql@columns@margins\else
3064       \ifdim\eql@tagmargin@>\z@
3065         \advance\eql@colsep@-\eql@tagsepmin@
3066       \fi
3067     \fi
3068   \fi

3069   \count@\eql@columns@inter@
3070   \ifdefined\eql@columns@margins
3071     \ifdefined\eql@layoutleft
3072       \advance\count@\@ne
3073     \else
3074       \advance\count@\tw@
3075     \fi
3076   \fi
3077   \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 (maybe some distinctions based on flush left)

```
3078   \ifdim\eql@colsep@<\eql@colsepmin@
3079     \eql@colsep@\eql@colsepmin@
3080   \else
```

```

3081 \dimen@glueexpr\eq@colsepmax@val\relax
3082 \ifdim\eq@colsep@>\dimen@
3083 \eq@colsep@\dimen@
3084 \fi
3085 \fi

```

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

```

3086 \ifdefined\eq@layoutleft
3087 \ifdim\eq@colsep@=\eq@colsepmin@
3088 \eq@marginleft@\displaywidth
3089 \advance\eq@marginleft@-\eq@totalwidth@
3090 \advance\eq@marginleft@-\eq@columns@inter@\eq@colsep@
3091 \ifdim\eq@marginleft@>\eq@layoutleftmargin@
3092 \eq@marginleft@\eq@layoutleftmargin@
3093 \else
3094 \ifdim\eq@marginleft@<\eq@layoutleftmarginmin@
3095 \eq@marginleft@\eq@layoutleftmarginmin@
3096 \fi
3097 \fi
3098 \else
3099 \eq@marginleft@\eq@layoutleftmargin@
3100 \fi
3101 \else
3102 \ifdefined\eq@columns@margins
3103 \eq@marginleft@\displaywidth
3104 \advance\eq@marginleft@-\eq@totalwidth@
3105 \advance\eq@marginleft@-\eq@columns@inter@\eq@colsep@
3106 \ifdim\eq@marginleft@<\eq@tagmargin@
3107 \eq@marginleft@\z@
3108 \else
3109 \advance\eq@marginleft@-\eq@tagmargin@
3110 \divide\eq@marginleft@\tw@
3111 \fi
3112 \ifdefined\eq@tagsleft
3113 \advance\eq@marginleft@\eq@tagmargin@
3114 \fi
3115 \else
3116 \ifdefined\eq@tagsleft
3117 \eq@marginleft@\eq@tagmargin@
3118 \ifdim\eq@tagmargin@>\z@
3119 \advance\eq@marginleft@\eq@tagsepmin@
3120 \fi
3121 \else
3122 \eq@marginleft@\z@
3123 \fi
3124 \fi
3125 \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:

```

3126 \ifdefined\eq@tagsleft
3127 \let\eq@columns@adjust@test\eq@columns@adjust@test@tagleft
3128 \else
3129 \let\eq@columns@adjust@test\eq@columns@adjust@test@tagright
3130 \fi
3131 \eq@row@\eq@totalrows@
3132 \loop\ifnum\eq@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:

```

3133 \ifnum\eq@numbering@target@<\z@
3134 \eq@tagwidth@\eq@tagwidth@get\eq@row@\relax
3135 \else
3136 \ifnum\eq@numbering@target@=\eq@row@
3137 \eq@tagwidth@\eq@tagwidth@get\z@\relax
3138 \fi
3139 \fi
3140 \ifdim\eq@tagwidth@>\z@
3141 \eq@columns@adjust@calc
3142 \eq@columns@adjust@test
3143 \fi
3144 \advance\eq@row@\m@ne
3145 \repeat

```

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

```

3146 \advance\eq@totalwidth@\eq@columns@inter@\eq@colsep@
3147 \advance\eq@totalwidth@\eq@marginleft@
3148 }

```

Calc Space.

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

```

3149 \def\eq@columns@adjust@calc{%
3150 \eq@line@pos@\z@
3151 \eq@column@\z@
3152 \eq@line@avail@\eq@totalwidth@
3153 \@tempcnta\eq@totalcolumns@
3154 \eq@line@width@\z@
3155 \@tempcntb\z@
3156 \edef\@tempb{\eq@fieldlength@get\eq@row@}%
3157 \@for\@tempa:=\@tempb\do
3158 \eq@columns@adjust@calc@col
3159 \advance\@tempcnta\m@ne
3160 \divide\@tempcnta\tw@
3161 \advance\@tempcntb\m@ne
3162 \divide\@tempcntb\tw@
3163 }

```

`\eq@columns@adjust@calc@col` The macro `\eq@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:

```

3164 \def\eq@columns@adjust@calc@col{%
3165 \advance\eq@column@\@ne
3166 \@tempdima\@tempa\relax

```



```

3167 \dimen@eql@columns@colwidth@get\eql@column@relax
3168 \ifdim@tempdima>\z@
3169 \ifdim\eql@line@width@=\z@
3170 \eql@line@avail@\eql@line@pos@
3171 \@tempcnta\eql@column@
3172 \ifodd\eql@column@
3173 \advance\eql@line@avail@\dimen@
3174 \advance\eql@line@avail@-\@tempdima
3175 \fi
3176 \fi
3177 \eql@line@width@\eql@line@pos@
3178 \@tempcntb\eql@column@
3179 \ifodd\eql@column@
3180 \advance\eql@line@width@\dimen@
3181 \else
3182 \advance\eql@line@width@\@tempdima
3183 \fi
3184 \fi
3185 \advance\eql@line@pos@\dimen@
3186 }

```

Placement for Right Tags.

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

```
3187 \def\eql@columns@adjust@test@tagright{%
```

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

```

3188 \@tempdima\displaywidth
3189 \advance\@tempdima-\eql@line@width@
3190 \advance\@tempdima-\eql@tagwidth@

```

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

```

3191 \dimen@\eql@marginleft@
3192 \advance\dimen@\@tempcntb\eql@colsep@
3193 \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).

```

3194 \dimen@\@tempcntb\eql@colsepmin@
3195 \ifdefined\eql@layoutleft
3196 \advance\dimen@\eql@layoutleftmarginmin@
3197 \fi
3198 \ifdim\dimen@>\@tempdima\else

```

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

```

3199 \eql@columns@adjust@modify@tagright
3200 \fi
3201 \fi
3202 }

```

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

```
3203 \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.

```

3204 \ifnum\@tempcntb>\z@
3205   \dimen@\@tempdima
3206   \count@\@tempcntb
3207   \ifdefined\eq@layoutleft
3208     \advance\dimen@-\eq@marginleft@
3209   \else
3210     \ifdefined\eq@columns@margins
3211       \advance\count@\@ne
3212     \fi
3213   \fi
3214   \divide\dimen@\count@

```

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

```

3215   \ifdim\dimen@<\eq@colsep@
3216     \ifdim\dimen@<\eq@colsepmin@
3217       \dimen@\eq@colsepmin@
3218     \fi
3219     \eq@colsep@\dimen@
3220   \fi
3221 \fi

```

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

```

3222 \dimen@\@tempdima
3223 \advance\dimen@-\@tempcntb\eq@colsep@
3224 \ifdim\dimen@<\eq@marginleft@
3225   \eq@marginleft@\dimen@
3226 \fi
3227 }

```

Placement for Left Tags.

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

```

3228 \def\eq@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.

```

3229 \count@\eq@columns@inter@
3230 \advance\count@-\@tempcnta
3231 \@tempdima\eq@tagwidth@
3232 \advance\@tempdima-\eq@line@avail@

```

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

```

3233 \dimen@\eq@marginleft@
3234 \advance\dimen@\@tempcnta\eq@colsep@
3235 \ifdim\dimen@<\@tempdima

```

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

```

3236   \ifdefined\eql@layoutleft
3237     \dimen@\eql@layoutleftmarginmax@
3238   \else
3239     \dimen@\displaywidth
3240   \fi
3241   \ifdim\dimen@>\eql@tagwidth@

```

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

```

3242     \dimen@\count@\eql@colsepmin@
3243     \advance\dimen@\eql@totalwidth@
3244     \advance\dimen@\@tempdima
3245     \ifdim\dimen@>\displaywidth\else

```

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

```

3246         \eql@columns@adjust@modify@tagleft
3247     \fi
3248   \fi
3249 \fi
3250 }

```

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?!

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

```

3251 \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.

```

3252   \ifnum\count@>\z@
3253     \dimen@\displaywidth
3254     \advance\dimen@-\eql@totalwidth@
3255     \advance\dimen@-\@tempdima
3256     \ifdefined\eql@columns@margins
3257       \advance\count@\@ne
3258     \fi
3259     \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.

```

3260     \ifdim\dimen@<\eql@colsep@
3261       \ifdim\dimen@<\eql@colsepmin@
3262         \dimen@\eql@colsepmin@
3263       \fi
3264       \advance\dimen@-\eql@colsep@
3265       \advance\eql@marginleft@-\eql@columns@inter@\dimen@
3266       \advance\eql@colsep@\dimen@
3267     \fi
3268   \fi

```

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

```

3269 \dimen@-\@tempcnta\eql@colsep@
3270 \advance\dimen@\@tempdima
3271 \ifdim\dimen@>\eql@marginleft@
3272   \eql@marginleft@\dimen@
3273 \fi
3274 }

```

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.

```
3275 \newtoks\eql@scan@reg@
```

`\eql@scan@body@dump` The macro `\eql@scan@body@dump` dumps the equation body from the register so that we do not have to pass it around in arguments. The macro `\eql@scan@body@rescan` rescans the tokens so that special commands such as `\verb` can be processed properly. The register `\eql@scan@body` holds the currently selected mode of operation:

```

3276 \def\eql@scan@body@dump{\the\eql@scan@reg@}
3277 \def\eql@scan@body@rescan{%
3278   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
3279 \let\eql@scan@body\eql@scan@body@dump

```

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

```
3280 \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@`:

```

3281 \def\eql@scan@env#1{%
3282   (dev)\eql@dev@enter\eql@scan@env
3283   \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenenv}}%
3284   \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`:

```

3285 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
3286 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
3287 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
3288 \eql@scan@iterate
3289 }

```

`\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:

```

3290 \def\eql@scan@env@iterate#1\end#2{%
3291 \edef\eql@scan@stack{%
3292 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
3293 \ifx\@empty\eql@scan@stack
3294 \@checkend{#2}%
3295 \eql@scan@addto{#1}%
3296 \expandafter\let\eql@scan@iterate\eql@scan@env@org
3297 (dev)\eql@dev@leave\eql@scan@env
3298 \expandafter\eql@scan@end
3299 \else
3300 \eql@scan@addto{#1\end{#2}}%
3301 \expandafter\eql@scan@iterate
3302 \fi
3303 }

```

`\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.

```

3304 \long\def\eql@scan@env@count#1\begin#2{%
3305 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
3306 }

```

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

```

3307 \def\eql@scan@env@cancel{%
3308 \@namedef{end\@currenvir}{\ignorespacesafterend}%
3309 }

```

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 `\]`:

```

3310 \def\eql@scan@sqr#1{%
3311 (dev)\eql@dev@enter\eql@scan@sqr
3312 \def\eql@scan@end{#1\]}%
3313 \eql@scan@reg@{\def\eql@scan@stack{b}%
3314 \let\eql@scan@sqr@org\[%\]
3315 \let\[%\eql@scan@sqr@iterate%\]
3316 \[%\]
3317 }

```

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

```

3318 \def\eql@scan@sqr@iterate#1\]{%
3319   \edef\eql@scan@stack{%
3320     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
3321   \ifx\@empty\eql@scan@stack
3322     \let\[\eql@scan@sqr@org%\]
3323     \eql@scan@addto{#1}%
3324 (dev)\eql@dev@leave\eql@scan@sqr
3325   \expandafter\eql@scan@end
3326   \else
3327     \eql@scan@addto{#1\]}%
3328     \expandafter\[%\]
3329   \fi
3330 }

```

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

```

3331 \long\def\eql@scan@sqr@count#1\[#2{\]
3332   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
3333 }

```

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

```

3334 \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 `\>`:

```

3335 \def\eql@scan@ang#1{%
3336 (dev)\eql@dev@enter\eql@scan@ang
3337   \def\eql@scan@end{#1\>}%
3338   \eql@scan@reg@{\def\eql@scan@stack{b}%
3339     \let\eql@scan@ang@org\<%\>
3340     \let\<\eql@scan@ang@iterate%\>
3341     \<%\>
3342 }

```

Iterate until we find a balanced pairing of angle brackets:

```

3343 \def\eql@scan@ang@iterate#1\>{%
3344   \edef\eql@scan@stack{%
3345     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
3346   \ifx\@empty\eql@scan@stack
3347     \let\<\eql@scan@ang@org%\>
3348     \eql@scan@addto{#1}%
3349 (dev)\eql@dev@leave\eql@scan@ang
3350   \expandafter\eql@scan@end
3351   \else
3352     \eql@scan@addto{#1\>}%
3353     \expandafter\<%\>
3354   \fi
3355 }

```

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

```

3356 \long\def\eql@scan@ang@count#1\<#2{\>
3357   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
3358 }

```

O.2 Options Processing

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

```
3359 \def\eql@equations@testall{\eql@equations@testtilde}
3360 \def\eql@equations@testtilde#1{%
3361   \eql@ifnextgobble@tight~%
3362   {\eqnaddopt{lines}\eql@equations@testopt{#1}}%
3363   {\eql@equations@testopt{#1}}}
3364 \def\eql@equations@testopt#1{%
3365   \eql@ifnextchar@tight[%]
3366   {\eql@equations@addopt{\eql@equations@testexcl{#1}}}%
3367   {\eql@equations@testexcl{#1}}}
3368 \def\eql@equations@addopt#1[#2]{\eqnaddopt{#2}#1}
3369 \def\eql@equations@testexcl#1{%
3370   \eql@ifnextgobble@tight!%
3371   {\eqnaddopt{donumber}\eql@equations@testat{#1}}}%
3372   {\eql@equations@teststar{#1}}}
3373 \def\eql@equations@teststar#1{%
3374   \eql@ifstar@tight%
3375   {\eqnaddopt{nonumber}\eql@equations@testat{#1}}}%
3376   {\eql@equations@testat{#1}}}
3377 \def\eql@equations@testat#1{%
3378   \eql@ifat@tight
3379   {\eql@equations@addlabel{#1}}}%
3380   {#1}}
3381 \def\eql@equations@addlabel#1#2{\eqnaddopt{label={#2}}#1}
```

`\eql@equations@processopt` The macro `\eql@equations@processopt` processes the options received by `\eqnaddopt`. First, clear several non-persistent registers (labels, tags, direct vertical spacing). Then process the arguments. Finally evaluate `\eql@indent@val` and `\eql@tagsepmin@val` and prevent main punctuation from being passed to nested environments:

```
3382 \def\eql@equations@processopt{%
3383   \let\eql@blocklabel\@undefined
3384   \let\eql@blocktag\@undefined
3385   \let\eql@skip@force@above\@undefined
3386   \let\eql@skip@force@below\@undefined
3387   \let\eql@skip@force@leave\@undefined
3388   \eql@abovespace@\z@skip
3389   \eql@belowspace@\z@skip
3390   \eql@displaybreak@prepen@\@MM
3391   \eql@nextopt@process{equations}%
3392   \let\eql@punct@block\eql@punct@main
3393   \let\eql@punct@main\relax
3394   \eql@indent@\glueexpr\eql@indent@val\relax
3395   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
3396 }
```

O.3 Single-Line Main

```
3397 \def\eql@single@cr{%
3398   \eql@error{Cannot use ‘\string\’ within display equation.
3399   Please switch to equations environment}}%
```

3400 }

TODO: describe

```
3401 \def\eq@single@start{%
3402   \eq@halign@catchprevdepth
3403   \eq@tagging@start
3404   \eq@dollar@begin
3405   \eq@numbering@eval@mode
3406   \let\eq@numbering@subeq@use\eq@false
3407   \eq@stack@save@single
```

TODO: make other display environments push these!?

```
3408   \eq@numbering@single@init
3409   \ifdefined\eq@single@native
3410     \let\eq@single@start@sel\eq@single@start@native
3411     \let\eq@single@end@sel\eq@single@end@native
3412     \let\raisetag\eq@raisetag@default
3413   \else
3414     \let\eq@single@start@sel\eq@single@start@print
3415     \let\eq@single@end@sel\eq@single@end@print
3416   \fi
3417   \ifdefined\eq@single@crerror\else
3418     \let\\eq@single@cr
3419   \fi
3420   \eq@single@start@sel
3421 }
```

```
3422 \def\eq@single@end{%
3423   \eq@punct@apply@block
3424   \eq@hook@eqout
3425   \eq@single@end@sel
3426   \eq@stack@restore
3427   \eq@dollar@end
3428   \eq@tagging@end
3429   \eq@halign@leave
3430 }
```

TODO: : try to feed in tagging after catchprevdepth

```
3431 \def\eq@single@main{%
3432   \expandafter\eq@single@start
3433   \eq@scan@body
3434   \eq@single@end
3435 }
```

TODO: describe

```
3436 \def\eq@mode@equation{%
3437   \ifdefined\eq@single@doscan
3438     \let\eq@equations@main\eq@single@main
3439     \let\eq@equations@end\@empty
3440   \else
3441     \let\eq@equations@main\@undefined
3442     \let\eq@equations@end\eq@single@end
3443   \fi
3444 }
```


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`

```
3445 \let\eql@multi@mode@lines\eql@false
3446 \def\eql@multi@main{%
3447   \eql@halign@catchprevdepth
3448   \eql@tagging@start
3449   \eql@dollar@dollar@begin
3450   \eql@numbering@eval@mode
3451   \eql@stack@save@multi
3452   \ifdefined\eql@subequations@active
3453     \let\eql@numbering@subeq@use\eql@false
3454   \fi
3455   \ifdefined\eql@numbering@subeq@use
3456     \eql@numbering@subeq@init
3457   \fi
3458   \let\intertext\eql@intertext
3459   \let\endintertext\endeql@intertext
3460   \let\shoveleft\eql@adjust@alignleft
3461   \let\shovecenter\eql@adjust@aligncenter
3462   \let\shoveright\eql@adjust@alignright
3463   \ifdefined\eql@multi@mode@lines
3464     \expandafter\eql@lines@measure
3465   \else
3466     \ifdefined\eql@ampproof@active
3467       \eql@ampproof
3468     \fi
3469     \expandafter\eql@columns@measure
3470   \fi
3471   \ifx\eql@numbering@subeq@use\@ne
3472     \eql@numbering@subeq@revert
3473   \fi
3474   \ifdefined\eql@multi@mode@lines\else
3475     \ifdefined\eql@multi@lines@fallback
3476       \ifnum\eql@totalcolumns@=\@ne
3477         \let\eql@multi@mode@lines\eql@true
3478       %
3479         \fi
3480     \fi
3481   \fi
3482   \ifdefined\eql@multi@mode@lines
3483     \expandafter\eql@lines@print
3484   \else
3485     \expandafter\eql@columns@print
3486   \fi
3487   \ifdefined\eql@numbering@subeq@use
3488     \eql@numbering@subeq@close
3489   \fi
3490   \eql@stack@restore
3491   \eql@dollar@dollar@end
3492   \eql@tagging@end
3493   \eql@halign@leave
3494 }
```

TODO: describe

```
3495 \def\eql@mode@columns{%
3496   \let\eql@equations@main\eql@multi@main
3497   \let\eql@equations@end\@empty
3498   \let\eql@multi@mode@lines\eql@false
3499 }
3500 \def\eql@mode@lines{%
3501   \let\eql@equations@main\eql@multi@main
3502   \let\eql@equations@end\@empty
3503   \let\eql@multi@mode@lines\eql@true
3504 }
```

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`:

```
3505 \newenvironment{equations}{%
3506 (dev)\eql@dev@enterenv
3507   \ifmmode
3508     \eql@error@mathmode{\string\begin{\@currenvir}}%
3509     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
3510   \else
3511     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3512     \expandafter\eql@equations@start
3513   \fi
3514 }{%
3515   \eql@equations@end
3516   \ignorespacesafterend
3517 (dev)\eql@dev@leaveenv
3518 }
```

`\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`:

```
3519 \def\eql@equations@start{%
3520   \eql@equations@processopt
3521   \ifdefined\eql@equations@main
3522     \expandafter\eql@scan@env\expandafter\eql@equations@main
3523   \else
3524     \expandafter\eql@single@start
3525   \fi
3526 }
```

Square Brackets.

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

```
3527 \newenvironment{equations@sqr}{}{}
```

`\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`:

```

3528 \protected\def\eql@equations@sqr@open{%
3529   \ifmmode
3530     \eql@error@mathmode{\string\[...\string\]}%
3531     \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
3532   \else
3533 (dev)\eql@dev@enter{\[...\string\]}%
3534   \expandafter\eqnadopt\expandafter{\eql@equations@sqr@opt}%
3535   \begin{equations@sqr}%
3536   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3537   \expandafter\eql@equations@sqr@start
3538   \fi
3539 }

```

`\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`:

```

3540 \def\eql@equations@sqr@start{%
3541   \eql@equations@processopt
3542   \ifdefined\eql@equations@main
3543     \expandafter\eql@scan@sqr\expandafter\eql@equations@main
3544   \else
3545     \expandafter\eql@single@start
3546   \fi
3547 }

```

`\equations@sqr@close` The macro `\eql@equations@sqr@close` holds the definition for ‘`\]`’.

```

3548 \protected\def\eql@equations@sqr@close{%
3549   \eql@equations@end
3550 (dev)\eql@dev@leave{\]...\string\]}%
3551   \end{equations@sqr}%
3552   \ignorespaces
3553 }

```

Angle Brackets.

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

```

3554 \newenvironment{equations@ang}{}{}

```

`\equations@ang@open` The macro `\eql@equations@ang@open` holds the definition for ‘`\langle`’.

```

3555 \protected\def\eql@equations@ang@open{%
3556   \ifmmode
3557     \eql@error@mathmode{\string\langle...\string\rangle}%
3558     \expandafter\eql@scan@ang\expandafter\eql@scan@sqrang@cancel
3559   \else
3560 (dev)\eql@dev@enter{\langle...\string\rangle}%
3561   \expandafter\eqnadopt\expandafter{\eql@equations@ang@opt}%
3562   \begin{equations@ang}%
3563   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3564   \expandafter\eql@equations@ang@start
3565   \fi
3566 }

```

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

```
3567 \def\eq@equations@ang@start{%
3568   \eq@equations@processopt
3569   \ifdefined\eq@equations@main
3570     \expandafter\eq@scan@ang\expandafter\eq@equations@main
3571   \else
3572     \expandafter\eq@single@start
3573   \fi
3574 }
```

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

```
3575 \protected\def\eq@equations@ang@close{%
3576   \eq@equations@end
3577   \dev\eq@dev@leave{\<...\string\>}%
3578   \end{equations@ang}%
3579   \ignorespaces
3580 }
```

P Options

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

```
3581 \RequirePackage{keyval}
```

P.1 Selection Tools

`\eq@decide@select` Some parameter values take values in a given set, e.g. `true` vs. `false` or `left` vs. `right`. The macro `\eq@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:

```
3582 \def\eq@decide@select#1#2#3#4{%
3583   \def\@tempa{#3}%
3584   \let\@tempd\undefined
3585   \@for\@tempc:=#4\do{%
3586     \ifdefined\@tempd\else
3587       \edef\@tempb{\noexpand\@tempb:=\expandafter\@firstoftwo\@tempc}%
3588       \expandafter\@for\@tempb\do{%
3589         \ifx\@tempa\@tempb
3590           \expandafter\expandafter\expandafter\def
3591             \expandafter\expandafter\expandafter\@tempd
3592               \expandafter\expandafter\expandafter{%
3593                 \expandafter\@secondoftwo\@tempc}%
3594         \fi
3595       }%
3596     \fi
3597   }%
3598   \ifdefined\@tempd
```

```

3599   \@tempd
3600   \else
3601     \eql@error{undefined value ‘#3’ for option ‘#2’ of ‘#1’}%
3602   \fi
3603 }

```

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

```

3604 \def\eql@decide@if#1#2#3#4#5{%
3605   \eql@decide@select{#1}{#2}{#3}{%
3606     {on,true,yes,enabled,1}{#4}},%
3607     {off,false,no,disabled,0}{#5}}}}

```

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

```

3608 \def\eql@decide@bool#1#2#3#4{%
3609   \eql@decide@if{#1}{#2}{#3}{\let#4\eql@true}{\let#4\eql@false}}

```

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

```

3610 \def\eql@decide@abovebelow#1#2#3#4#5{%
3611   \eql@decide@select{#1}{#2}{#3}{%
3612     {,abovebelow,both,tb}{#4#5},%
3613     {above,top,t}{#4},%
3614     {below,bottom,b}{#5}}}}

```

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

```

3615 \def\eql@decide@situation#1#2#3#4{%
3616   \eql@decide@select{#1}{#2}{#3}{%
3617     {{long}{\def#4{0}}},%
3618     {{short}{\def#4{1}}},%
3619     {{cont}{\def#4{2}}},%
3620     {{par}{\def#4{3}}},%
3621     {{top}{\def#4{4}}},%
3622     {{noskip}{\def#4{5}}},%
3623     {{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:

```

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

```

```

3638     \noexpand\eqL@for@fn{##1}{\eqL@for@varb}{\eqL@for@vara}}%
3639     \eqL@for@call{##1}%
3640   }%
3641 }%
3642 }

```

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

```

3643 \def\eqL@setkeys#1#2{%
3644   \def\eqL@tmp{\setkeys{eqL@#1}}%
3645   \expandafter\eqL@tmp\expandafter{#2}%
3646 }

```

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

```

\eqnaddopt
3647 \let\eqL@nextopt\@empty
3648 \def\eqL@nextopt@process#1{%
3649 (dev)\eqL@dev@start\eqL@nextopt@process
3650   \eqL@setkeys{#1}\eqL@nextopt
3651   \let\eqL@tagging@opt\eqL@nextopt
3652   \global\let\eqL@nextopt\@empty
3653 }
3654 \newcommand{\eqnaddopt}[1]{%
3655   \expandafter\def\expandafter\eqL@nextopt\expandafter{\eqL@nextopt,#1}}

```

P.3 Options Declarations

TODO: describe

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

```

3656 \eqL@define@key{equationsbox}{gathered,gather,ga,lines,ln,\string~}[]{%
3657   \eqL@mode@stacked}
3658 \eqL@define@key{equationsbox}{aligned,align,al,columns,col,@}[]{%
3659   \eqL@mode@aligned}
3660 \eqL@define@key{equationsbox}{top,t}[]{\let\eqL@box@box\vtop}
3661 \eqL@define@key{equationsbox}{center,c}[]{\let\eqL@box@box\vcenter}
3662 \eqL@define@key{equationsbox}{bottom,b}[]{\let\eqL@box@box\ vbox}
3663 \eqL@define@key{equationsbox}{colsep}{\def\eqL@box@colsep{#1}}

```

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

```

3664 \eqL@define@key{equations}{equation,eq,single,1}[]{\eqL@mode@equation}
3665 \eqL@define@key{equations}{gathered,gather,ga,lines,ln,\string~}[]{%
3666   \eqL@mode@lines}
3667 \eqL@define@key{equations}{aligned,align,al,columns,col,@}[]{%
3668   \eqL@mode@columns}
3669 \eqL@define@key{equations}{native}[true]{%
3670   \eqL@decide@bool{#3}{#2}{#1}\eqL@single@native%
3671   \ifdefined\eqL@single@native\let\eqL@layoutleft\eqL@false\fi}
3672 \eqL@define@key{setup}{native}[true]{%
3673   \eqL@decide@bool{#3}{#2}{#1}\eqL@single@native}
3674 \eqL@define@key{setup}{scanequation}[true]{%
3675   \eqL@decide@bool{#3}{#2}{#1}\eqL@single@doscan}
3676 \eqL@define@key{setup}{sqropt}[]{%

```

```

3677 \def\eqlequations@sqr@opt{equation,#1}
3678 \eql@define@key{setup}{angopt}[]{%
3679 \def\eqlequations@ang@opt{align,#1}}

```

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

```

3680 \def\eql@keycat{equations,equationsbox,setup}
3681 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
3682 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@if{#3}{#2}{#1}%
3683 {\let\eql@strut@field\eql@strut}{\let\eql@strut@field\relax}}
3684 \eql@define@key\eql@keycat{strutttag}[true]{\eql@decide@if{#3}{#2}{#1}%
3685 {\let\eql@strut@tag\eql@strut}{\let\eql@strut@tag\relax}}

```

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

```

3686 \eql@define@key{equations}{displaybreak}[4]{\eql@displaybreak@pre{#1}}
3687 \def\eql@keycat{equations,setup}
3688 \eql@define@key\eql@keycat{allowbreaks,allowdisplaybreaks}[4]{%
3689 \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax}
3690 \eql@define@key\eql@keycat{displayheight}[\ht\eql@strutbox@]{%
3691 \def\eql@display@height{#1}}
3692 \eql@define@key\eql@keycat{displaydepth}[\dp\eql@strutbox@]{%
3693 \def\eql@display@depth{#1}}
3694 \eql@define@key\eql@keycat{displayheight*}[]{%
3695 \let\eql@display@height\undefined}
3696 \eql@define@key\eql@keycat{displaydepth*}[]{%
3697 \let\eql@display@depth\undefined}

```

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

```

3698 \eql@define@key{equations}{noskip}[]{%
3699 \eql@decide@abovebelow{#3}{#2}{#1}%
3700 {\def\eql@skip@force@above{5}}%
3701 {\def\eql@skip@force@below{5}}%
3702 \eql@define@key{equations}{short}[above]{%
3703 \eql@decide@abovebelow{#3}{#2}{#1}%
3704 {\def\eql@skip@force@above{1}}%
3705 {\def\eql@skip@force@below{1}}%
3706 \eql@define@key{equations}{long}[]{%
3707 \eql@decide@abovebelow{#3}{#2}{#1}%
3708 {\def\eql@skip@force@above{0}}%
3709 {\def\eql@skip@force@below{0}}%
3710 \eql@define@key{equations}{medskip}[]{%
3711 \eql@decide@abovebelow{#3}{#2}{#1}%
3712 {\def\eql@skip@force@above{6}}%
3713 {\def\eql@skip@force@below{6}}%
3714 \eql@define@key{equations}{par}[par]{%
3715 \eql@decide@select{#3}{#2}{#1}{%
3716 {{default,}{\let\eql@skip@force@leave\undefined}},%
3717 {{cont,hmode}{\let\eql@skip@force@leave\z@}},%
3718 {{par,vmode}{\let\eql@skip@force@leave\one
3719 \ifdefined\eql@skip@force@below\else
3720 \def\eql@skip@force@below{3}%
3721 \fi}},%
3722 {{top}{\let\eql@skip@force@leave\tw@
3723 \ifdefined\eql@skip@force@below\else
3724 \def\eql@skip@force@below{4}
3725 \fi}}}}

```

TODO: describe

```
3726 \eql@define@key{equations}{skip}{%
3727   \def\eq@skip@force@above{7}%
3728   \def\eq@skip@custom@above{#1}%
3729   \let\eq@skip@force@below\eq@skip@force@above
3730   \let\eq@skip@custom@below\eq@skip@custom@above}
3731 \eql@define@key{equations}{aboveskip}{%
3732   \def\eq@skip@force@above{7}%
3733   \def\eq@skip@custom@above{#1}}
3734 \eql@define@key{equations}{belowskip}{%
3735   \def\eq@skip@force@below{7}%
3736   \def\eq@skip@custom@below{#1}}
3737 \eql@define@key{equations}{abovespace}{%
3738   \advance\eq@abovespace@glueexpr#1\relax}
3739 \eql@define@key{equations}{belowspace}{%
3740   \advance\eq@belowspace@glueexpr#1\relax}
```

TODO: describe

```
3741 \eql@define@key{intertext}{skip}{%
3742   \def\eq@skip@force@above{7}%
3743   \def\eq@skip@custom@above{#1}%
3744   \let\eq@skip@force@below\eq@skip@force@above
3745   \let\eq@skip@custom@below\eq@skip@custom@above}
3746 \eql@define@key{intertext}{aboveskip}{%
3747   \def\eq@skip@force@below{7}%
3748   \def\eq@skip@custom@below{#1}}
3749 \eql@define@key{intertext}{belowskip}{%
3750   \def\eq@skip@force@above{7}%
3751   \def\eq@skip@custom@above{#1}}
3752 \eql@define@key{intertext}{noskip}[]{%
3753   \eql@decide@abovebelow{#3}{#2}{#1}%
3754   {\def\eq@skip@force@below{5}}%
3755   {\def\eq@skip@force@above{5}}}
3756 \eql@define@key{intertext}{short}[]{%
3757   \eql@decide@abovebelow{#3}{#2}{#1}%
3758   {\def\eq@skip@force@below{1}}%
3759   {\def\eq@skip@force@above{1}}}
3760 \eql@define@key{intertext}{long}[]{%
3761   \eql@decide@abovebelow{#3}{#2}{#1}%
3762   {\def\eq@skip@force@below{0}}%
3763   {\def\eq@skip@force@above{0}}}
3764 \eql@define@key{intertext}{medskip}[]{%
3765   \eql@decide@abovebelow{#3}{#2}{#1}%
3766   {\def\eq@skip@force@below{6}}%
3767   {\def\eq@skip@force@above{6}}}
```

TODO: describe

```
3768 \eql@define@key{setup}{skip,longskip}{%
3769   \abovedisplayskip\glueexpr#1\relax
3770   \belowdisplayskip\abovedisplayskip
3771   \def\eq@skip@long@above{#1}%
3772   \let\eq@skip@long@below\eq@skip@long@above}
3773 \eql@define@key{setup}{aboveskip,abovelongskip}{%
3774   \abovedisplayskip\glueexpr#1\relax
3775   \def\eq@skip@long@above{#1}}
3776 \eql@define@key{setup}{belowskip,belowlongskip}{%
3777   \belowdisplayskip\glueexpr#1\relax}
```



```

3778 \def\eq@skip@long@below{#1}
3779 \eq@define@key{setup}{aboveshortskip}{%
3780 \abovedisplayshortskip\glueexpr#1\relax
3781 \def\eq@skip@short@above{#1}
3782 \eq@define@key{setup}{belowshortskip}{%
3783 \belowdisplayshortskip\glueexpr#1\relax
3784 \def\eq@skip@short@below{#1}
3785 \eq@define@key{setup}{tagskip}{%
3786 \def\eq@skip@tag@above{#1}%
3787 \let\eq@skip@tag@below\eq@skip@tag@above}
3788 \eq@define@key{setup}{abovetagskip}{%
3789 \def\eq@skip@tag@above{#1}
3790 \eq@define@key{setup}{belowtagskip}{%
3791 \def\eq@skip@tag@below{#1}
3792 \eq@define@key{setup}{medskip}{%
3793 \def\eq@skip@med@above{#1}%
3794 \let\eq@skip@med@below\eq@skip@med@above}
3795 \eq@define@key{setup}{abovemedskip}{%
3796 \def\eq@skip@med@above{#1}
3797 \eq@define@key{setup}{belowmedskip}{%
3798 \def\eq@skip@med@below{#1}
3799 \eq@define@key{setup}{medtagskip}{%
3800 \def\eq@skip@medtag@above{#1}%
3801 \let\eq@skip@medtag@below\eq@skip@medtag@above}
3802 \eq@define@key{setup}{abovemedtagskip}{%
3803 \def\eq@skip@medtag@above{#1}
3804 \eq@define@key{setup}{belowmedtagskip}{%
3805 \def\eq@skip@medtag@below{#1}
3806 \eq@define@key{setup}{abovetopskip}{%
3807 \def\eq@skip@top@above{#1}
3808 \eq@define@key{setup}{belowtopskip}{%
3809 \def\eq@skip@top@below{#1}
3810 \eq@define@key{setup}{aboveparskip}{%
3811 \def\eq@skip@par@above{#1}
3812 \eq@define@key{setup}{belowparskip}{%
3813 \def\eq@skip@par@below{#1}
3814 \eq@define@key{setup}{abovepartagskip}{%
3815 \def\eq@skip@partag@above{#1}
3816 \eq@define@key{setup}{belowpartagskip}{%
3817 \def\eq@skip@partag@below{#1}
3818 \eq@define@key{setup}{abovecontskip}{%
3819 \def\eq@skip@cont@above{#1}
3820 \eq@define@key{setup}{abovecontskip*}[]{%
3821 \def\eq@skip@cont@above{\eq@spread@val-\eq@skip@long@below}}
3822 \eq@define@key{setup}{belowcontskip}{%
3823 \def\eq@skip@cont@below{#1}
3824 \eq@define@key{setup}{shortmode}{%
3825 \eq@decide@select{#3}{#2}{#1}{%
3826 {{off,never,no}{\def\eq@skip@mode@short{0}}},%
3827 {{above,neverbelow,notbelow,belowoff}{\def\eq@skip@mode@short{1}}},%
3828 {{belowone,belowsingle}{\def\eq@skip@mode@short{2}}},%
3829 {{belowall,always,on}{\def\eq@skip@mode@short{3}}}}
3830 \eq@define@key{setup}{abovecontmode}{%
3831 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
3832 \eq@define@key{setup}{belowcontmode}{%
3833 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
3834 \eq@define@key{setup}{aboveparmode}{%
3835 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@above}

```

```

3836 \eql@define@key{setup}{belowparmode}{%
3837   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
3838 \eql@define@key{setup}{abovetopmode}{%
3839   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
3840 \eql@define@key{setup}{belowtopmode}{%
3841   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

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

```

3842 \def\eql@keycat{equations,subequations}
3843 \eql@define@key\eql@keycat{label}{\eql@blocklabel@set{#1}}
3844 \eql@define@key\eql@keycat{tag}{\eql@blocktag@set{#1}}
3845 \eql@define@key\eql@keycat{tag*}{\eql@blocktag@setstar{#1}}

```

Tag Spacing. **TODO:** describe

```

3846 \def\eql@keycat{equations,setup}
3847 \eql@define@key\eql@keycat{tagmargin}[]{\def\eql@tagmargin@val{#1}
3848   \ifx\eql@tagmargin@val@empty\let\eql@tagmargin@val@undefined\fi}
3849 \eql@define@key\eql@keycat{tagmargin*}{%
3850   \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}}
3851 \eql@define@key\eql@keycat{tagmarginratio}{%
3852   \eql@tagmargin@ratio@{dimexpr#1pt\relax}}
3853 \eql@define@key\eql@keycat{tagmarginthreshold}{%
3854   \def\eql@tagmargin@threshold{#1}}
3855 \eql@define@key\eql@keycat{mintagsep}{\def\eql@tagsepmin@val{#1}}
3856 \eql@define@key\eql@keycat{mintagwidth}{%
3857   \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}}
3858 \eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin@{#1}}

```

Tag Layout. **TODO:** describe

```

3859 \eql@define@key{setup}{tagbox,taglayout}{\eql@tag@setbox{#1}}
3860 \eql@define@key{setup}{tagbox,taglayout*}{\eql@tag@setbox@{#1}}
3861 \eql@define@key{setup}{tagform}{\eql@tag@setform#1}
3862 \eql@define@key{setup}{tagform*}{\eql@tag@setform@{#1}}
3863 \eql@define@key{setup}{subeqtemplate}{%
3864   \def\eql@subequations@template####1###2{#1}%
3865   \expandafter\def\expandafter\eql@subequations@template\expandafter{%
3866     \eql@subequations@template\theparentequation{equation}}%
3867 }
3868 %\def\eql@subequations@template{#1}}
3869 \eql@define@key{setup}{autolabel}[true]{%
3870   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@autolabel}
3871 \eql@define@key{setup}{autotag}[true]{%
3872   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@autotag}

```

Equation Numbering. **TODO:** describe

```

3873 \def\eql@keycat{equations,setup}
3874 \eql@define@key\eql@keycat{numberline,numline,n}[all]{%
3875   \eql@numbering@set{##1}}
3876 \eql@define@key\eql@keycat{nonumber,nn,*}[]{%
3877   \let\eql@numbering@active\eql@false}
3878 \eql@define@key\eql@keycat{donumber,dn,!}[]{%
3879   \let\eql@numbering@active\eql@true}
3880 \eql@define@key\eql@keycat{number,num}[true]{%

```

```

3881 \eql@decide@bool{#3}{#2}{#1}\eql@numbering@active}
3882 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
3883 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
3884 \eql@define@key\eql@keycat{tags,eqno}{%
3885 \eql@decide@select{#3}{#2}{#1}{%
3886   {{right,r}{\let\eql@tagsleft\eql@false}},%
3887   {{left,l}{\let\eql@tagsleft\eql@true}}}}

```

Horizontal Layout. **TODO:** describe

```

3888 \def\eql@keycat{equations,setup}
3889 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
3890   {{center,c}{\let\eql@layoutleft\eql@false}},%
3891   {{left,l}{\let\eql@layoutleft\eql@true}}}}
3892 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
3893 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
3894 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin@val{#1}}
3895 \eql@define@key\eql@keycat{leftmargin*}{%
3896   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin@val{\the\dimen@}}
3897 \eql@define@key\eql@keycat{minleftmargin}{%
3898   \eql@layoutleftmarginmin@\glueexpr#1\relax}
3899 \eql@define@key\eql@keycat{maxleftmargin}{%
3900   \eql@layoutleftmarginmax@\glueexpr#1\relax}
3901 \eql@define@key\eql@keycat{maxleftmargin*}[]{%
3902   \eql@layoutleftmarginmax@.5\maxdimen}

```

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

```

3903 \def\eql@keycat{equations,setup}
3904 \eql@define@key\eql@keycat{marginbadness}{\eql@marginbadness@#1\relax}
3905 \eql@define@key\eql@keycat{maxbadness}{\eql@maxbadness@#1\relax}
3906 \eql@define@key\eql@keycat{mincolsep}{\def\eql@colsepmin@val{#1}}
3907 \eql@define@key\eql@keycat{maxcolsep}{\def\eql@colsepmax@val{#1}}
3908 \eql@define@key\eql@keycat{maxcolsep*}[]{\def\eql@colsepmax@val{.5\maxdimen}}
3909 \eql@define@key\eql@keycat{margins}[true]{%
3910   \eql@decide@bool{#3}{#2}{#1}\eql@columns@margins}
3911 \def\eql@keycat{equationsbox,setup}
3912 \eql@define@key\eql@keycat{margin}{%
3913   \def\eql@box@marginleft{#1}\def\eql@box@marginright{#1}}
3914 \eql@define@key\eql@keycat{marginleft}{\def\eql@box@marginleft{#1}}
3915 \eql@define@key\eql@keycat{marginright}{\def\eql@box@marginright{#1}}

```

Horizontal Shape. **TODO:** describe

```

3916 \def\eql@keycat{equations,equationsbox,setup}
3917 \eql@define@key\eql@keycat{shape}[default]{\eql@shape@set{#1}}
3918 \eql@define@key\eql@keycat{padding,pad}[\eql@indent@val]{%
3919   \let\eql@paddingmax\eql@false
3920   \def\eql@paddingleft@val{#1}\def\eql@paddingright@val{#1}}
3921 \eql@define@key\eql@keycat{padleft}[\eql@indent@val]{%
3922   \let\eql@paddingmax\eql@false\def\eql@paddingleft@val{#1}}
3923 \eql@define@key\eql@keycat{padright}[\eql@indent@val]{%
3924   \let\eql@paddingmax\eql@false\def\eql@paddingright@val{#1}}
3925 \eql@define@key\eql@keycat{padmax}[true]{%
3926   \eql@decide@bool{#3}{#2}{#1}\eql@paddingmax}
3927 \eql@define@key\eql@keycat{indent}[2em]{%
3928   \def\eql@indent@val{#1}}

```

```

3929 \eql@define@key\eql@keycat{indent*}[2em]{%
3930   \def\eql@indent@val{#1}\def\eql@paddingleft@val{#1}}

```

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

```

3931 \def\eql@keycat{equations,equationsbox,setup}
3932 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
3933 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
3934 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
3935 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
3936 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
3937 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}{%
3938   {{ampeq,amprel,eqafter,beforerel}\eql@class@ampeq},%
3939   {{eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}}

```

Punctuation. **TODO:** describe

```

3940 \let\eql@punct@main\relax
3941 \def\eql@keycat{equations,equationsbox,setup}
3942 \eql@define@key\eql@keycat{punctsep}[\,]{\def\eql@punct@sep{#1}}
3943 \eql@define@key\eql@keycat{punct}[.]{\def\eql@punct@main{#1}}
3944 \eql@define@key\eql@keycat{punctline}[,]{\def\eql@punct@line{#1}}
3945 \eql@define@key\eql@keycat{punctcol}[,]{\def\eql@punct@col{#1}}
3946 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
3947 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
3948 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}

```

Global Switches. **TODO:** describe

```

3949 \let\eql@multi@linesfallback\eql@true
3950 \let\eql@single@crerror\eql@true
3951 \let\eql@ampproof@active\eql@false
3952 \eql@define@key{setup}{linesfallback}[true]{%
3953   \eql@decide@bool{#3}{#2}{#1}\eql@multi@linesfallback}
3954 \eql@define@key{setup}{ampproof}[true]{%
3955   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
3956 \eql@define@key{setup}{crerror}[true]{%
3957   \eql@decide@bool{#3}{#2}{#1}\eql@single@crerror}
3958 \eql@define@key{equations,setup}{rescan}[true]{%
3959   \eql@decide@if{#3}{#2}{#1}%
3960   {\let\eql@scan@body\eql@scan@body@rescan}%
3961   {\let\eql@scan@body\eql@scan@body@dump}}
3962 \eql@define@key{setup}{defaults}{%
3963   \eql@decide@select{#3}{#2}{#1}{%
3964     {{classic}{\eql@defaults@classic}},%
3965     {{eqnlines}{\eql@defaults@eqnlines}}}}

```

Package Options. **TODO:** describe

```

3966 \let\eql@provide@opt@equation\eql@true
3967 \let\eql@provide@opt@amsmathends\eql@true
3968 \let\eql@provide@opt@backup\eql@false
3969 \let\eql@provide@opt@amsmath\eql@true
3970 \let\eql@provide@opt@ang\eql@true
3971 \let\eql@provide@opt@eqref\eql@true
3972 \eql@define@key{setup}{equation}[true]{%
3973   \eql@error@packageoption{#2}%

```

```

3974 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@equation}
3975 \eql@define@key{setup}{amsmathends}[true]{%
3976 \eql@error@packageoption{#2}%
3977 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
3978 \eql@define@key{setup}{backup}[true]{%
3979 \eql@error@packageoption{#2}%
3980 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
3981 \eql@define@key{setup}{amsmath}[true]{%
3982 \eql@error@packageoption{#2}%
3983 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmath}
3984 \eql@define@key{setup}{ang}[true]{%
3985 \eql@error@packageoption{#2}%
3986 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
3987 \eql@define@key{setup}{eqref}[true]{%
3988 \eql@error@packageoption{#2}%
3989 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

P.4 Parameter Sets

TODO: par@above plus parskip?

```

3990 \def\eql@defaults@classic{%
3991 \eqnlineset{mintagsep={.5\fontdimen6\textfont2\relax}}%
3992 \eqnlineset{maxcolsep*}%
3993 \eqnlineset{spread={\jot}}%
3994 \eqnlineset{tagmargin}%
3995 \eqnlineset{tagmarginratio=1}%
3996 \eqnlineset{tagmarginthreshold=0.5}%
3997 \eqnlineset{leftmargin={\leftmargini}}%
3998 \eqnlineset{displayheight*}%
3999 \eqnlineset{displaydepth*}%
4000 \eqnlineset{shortmode=belowsingle}%
4001 \eqnlineset{abovecontmode=short}%
4002 \eqnlineset{belowcontmode=short}%
4003 \eqnlineset{aboveparmode=long}%
4004 \eqnlineset{belowparmode=long}%
4005 \eqnlineset{abovetopmode=long}%
4006 \eqnlineset{belowntopmode=long}%
4007 \eqnlineset{abovelongskip={\abovedisplayskip}}%
4008 \eqnlineset{belowlongskip={\belowdisplayskip}}%
4009 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
4010 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
4011 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
4012 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
4013 \eqnlineset{abovecontskip=0pt}%
4014 \eqnlineset{belowcontskip=0pt}%
4015 \eqnlineset{aboveparskip=0pt}%
4016 \eqnlineset{belowparskip=0pt}%
4017 \eqnlineset{abovetopskip=0pt}%
4018 \eqnlineset{belowntopskip=0pt}%
4019 \eqnlineset{abovetagskip=0pt}%
4020 \eqnlineset{belowntagskip=0pt}%
4021 \eqnlineset{abovemedtagskip=0pt}%
4022 \eqnlineset{belowmedtagskip=0pt}%
4023 \eqnlineset{abovepartagskip=0pt}%
4024 \eqnlineset{belowpartagskip=0pt}%
4025 }

```

values based on 10pt vs 12pt

```
4026 \def\eql@defaults@eqnlines{%
4027   \eqnlineset{mintagsep=.5em}%
4028   \eqnlineset{maxcolsep=2em}%
4029   \eqnlineset{spread={0.2\normalbaselineskip}}%
4030   \eqnlineset{tagmargin}%
4031   \eqnlineset{tagmarginratio=.334}%
4032   \eqnlineset{tagmarginthreshold=0.5}%
4033   \eqnlineset{leftmargin={\leftmargini}}%
4034   \eqnlineset{displayheight}%
4035   \eqnlineset{displaydepth}%
4036   \eqnlineset{shortmode=above}%
4037   \eqnlineset{abovecontmode=noskip}%
4038   \eqnlineset{belowcontmode=long}%
4039   \eqnlineset{aboveparmode=long}%
4040   \eqnlineset{belowparmode=long}%
4041   \eqnlineset{abovetopmode=noskip}%
4042   \eqnlineset{belowtopmode=long}%
4043   \eqnlineset{longskip={0.75\normalbaselineskip
4044     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
4045   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
4046     plus 0.25\normalbaselineskip}}%
4047   \eqnlineset{belowshortskip={0.0\normalbaselineskip
4048     plus 0.25\normalbaselineskip}}%
4049   \eqnlineset{medskip={0.4\normalbaselineskip
4050     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
4051   \eqnlineset{abovecontskip=0pt}%
4052   \eqnlineset{belowcontskip=0pt}%
4053   \eqnlineset{aboveparskip=0pt}%
4054   \eqnlineset{belowparskip=0pt}%
4055   \eqnlineset{abovetopskip=0pt}%
4056   \eqnlineset{belowtopskip=0pt}%
4057   \eqnlineset{abovetagskip=0pt}%
4058   \eqnlineset{belowtagskip=0pt}%
4059   \eqnlineset{abovemedtagskip=0pt}%
4060   \eqnlineset{belowmedtagskip=0pt}%
4061   \eqnlineset{abovepartagskip=0pt}%
4062   \eqnlineset{belowpartagskip=0pt}%
4063 }
```

P.5 Component Selection

TODO: describe

```
4064 \newenvironment{eql@gathered}
4065   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
4066 \newenvironment{eql@multlined}
4067   {\eqnaddopt{lines, padding, shape=steps}\equationsbox}{\endequationsbox}
4068 \newenvironment{eql@aligned}
4069   {\eqnaddopt{align}\equationsbox}{\endequationsbox}
```

TODO: describe

```
4070 \newenvironment{eql@equation}
4071   {\eqnaddopt{equation}\equations}{\endequations}
4072 \newenvironment{eql@gather}
4073   {\eqnaddopt{lines}\equations}{\endequations}
4074 \newenvironment{eql@multline}
```

```

4075 {\eqnadopt{lines,padmax,shape=steps,numberline=out}\equations}
4076 {\endequations}
4077 \newenvironment{eql@align}
4078 {\eqnadopt{align}\equations}{\endequations}
4079 \newenvironment{eql@flalign}
4080 {\eqnadopt{align,margins=false,maxcolsep*}\equations}{\endequations}
4081 \newenvironment{eql@equation*}
4082 {\eqnadopt{nonumber}\eql@equation}{\endequations}
4083 \newenvironment{eql@gather*}
4084 {\eqnadopt{nonumber}\eql@gather}{\endequations}
4085 \newenvironment{eql@multline*}
4086 {\eqnadopt{nonumber}\eql@multline}{\endequations}
4087 \newenvironment{eql@align*}
4088 {\eqnadopt{nonumber}\eql@align}{\endequations}
4089 \newenvironment{eql@flalign*}
4090 {\eqnadopt{nonumber}\eql@flalign}{\endequations}

```

TODO: describe

```

4091 \def\eql@provide@movecmd#1#2{%
4092   \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
4093 }
4094 \def\eql@provide@undefinecmd#1{%
4095   \expandafter\let\csname #1\endcsname\@undefined
4096 }
4097 \def\eql@provide@moveenv#1#2{%
4098   \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
4099   \expandafter\let\csname end#1\expandafter\endcsname\csname end#2\endcsname
4100 }
4101 \def\eql@provide@undefineenv#1{%
4102   \expandafter\let\csname #1\endcsname\@undefined
4103   \expandafter\let\csname end#1\endcsname\@undefined
4104 }

```

TODO: describe

```

4105 \def\eql@provide@onlyonce#1#2{%
4106   \def\eql@tmp{#2}%
4107   \def\@tempa{#1}%
4108   \ifx\eql@tmp\@tempa
4109     \let\eql@tmp\@undefined
4110   \fi
4111   \ifx\eql@tmp\@empty
4112     \let\eql@tmp\@undefined
4113   \fi
4114   \def\@tempa{*}%
4115   \ifx\eql@tmp\@tempa
4116     \def\eql@tmp{#1}%
4117   \fi
4118   \ifdefined\eql@tmp\else
4119     \ifcsname eql@provided@#1\endcsname
4120       \def\eql@tmp{#1}%
4121     \else
4122       \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4123     \fi
4124   \fi
4125 }

```

TODO: describe

```

4126 \def\eql@provide@cmdonlyonce#1#2{%

```

```

4127 \def\eql@tmp{#2}%
4128 \edef\@tempb{\expandafter\noexpand\csname#1\endcsname}%
4129 \ifx\eql@tmp\@tempb
4130   \let\eql@tmp\undefined
4131 \fi
4132 \ifx\eql@tmp\@empty
4133   \let\eql@tmp\undefined
4134 \fi
4135 \def\@tempa{*}%
4136 \ifx\eql@tmp\@tempa
4137   \let\eql@tmp\@tempb
4138 \fi
4139 \ifdefined\eql@tmp
4140   \edef\eql@tmp{\expandafter\expandafter\expandafter\@gobble
4141     \expandafter\string\eql@tmp}%
4142 \else
4143   \ifcsname eql@provided@#1\endcsname
4144     \let\eql@tmp\@tempb
4145   \else
4146     \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4147   \fi
4148 \fi
4149 }

```

TODO: describe

```

4150 \def\eql@provide@cmd#1#2{%
4151   \eql@provide@cmdonlyonce{#1}{#2}%
4152   \ifdefined\eql@tmp
4153     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@#1}%
4154   \else
4155     \eql@amsmath@after{%
4156       \eql@provide@movecmd{ams#1}{#1}%
4157       \eql@provide@movecmd{#1}{eql@#1}%
4158     }%
4159     \AddToHook{package/mathtools/after}{%
4160       \eql@provide@movecmd{#1}{eql@#1}%
4161     }%
4162     \eql@provide@movecmd{#1}{eql@#1}%
4163     \eql@amsmath@before{\eql@provide@undefinecmd{#1}}%
4164   \fi
4165 }

```

TODO: describe

```

4166 \def\eql@amsmath@endfix#1#2{%
4167   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
4168   \long\edef\@tempb{\expandafter\noexpand\csname eql@amsmath@end#2\endcsname}%
4169   \expandafter\ifx\csname end#1\endcsname\@tempa
4170     \expandafter\let\csname end#1\endcsname\@tempb
4171   \fi
4172 }

```

TODO: describe

```

4173 \def\eql@amsmath@fixends{%
4174   \eql@amsmath@after{%
4175     \let\eql@amsmath@endmultiline\endmultiline
4176     \eql@amsmath@endfix{multiline*}{multiline}%
4177     \let\eql@amsmath@endgather\endgather
4178     \eql@amsmath@endfix{gather*}{gather}%

```



```

4179 \let\eqL@amsmath@endalign\endalign
4180 \eqL@amsmath@endfix{align*}{align}%
4181 \eqL@amsmath@endfix{flalign}{align}%
4182 \eqL@amsmath@endfix{flalign*}{align}%
4183 \eqL@amsmath@endfix{alignat}{align}%
4184 \eqL@amsmath@endfix{alignat*}{align}%
4185 \eqL@amsmath@endfix{xalignat}{align}%
4186 \eqL@amsmath@endfix{xalignat*}{align}%
4187 \eqL@amsmath@endfix{xxalignat}{align}%
4188 \let\eqL@amsmath@endaligned\endaligned
4189 \eqL@amsmath@endfix{gathered}{aligned}%
4190 \eqL@amsmath@endfix{alignedat}{aligned}%
4191 }
4192 }

```

TODO: describe

```

4193 \def\eqL@backup@env@amsmath#1{%
4194 \eqL@amsmath@after{%
4195 \eqL@provide@moveenv{ams#1}{#1}%
4196 \eqL@provide@moveenv{ams#1*}{#1*}%
4197 }%
4198 }

```

TODO: describe

```

4199 \def\eqL@backup@env@equation{%
4200 \eqL@amsmath@after{%
4201 \eqL@provide@moveenv{amsequation}{equation}%
4202 \eqL@tagging@register@env{amsequation}%
4203 \eqL@provide@moveenv{amsequation*}{equation*}%
4204 \eqL@tagging@register@env{amsequation*}%
4205 }%
4206 \AddToHook{package/hyperref/after}{%
4207 \@ifpackageloaded{amsmath}{}{%
4208 \let\latexequation\H@equation
4209 \let\endlatexequation\H@endequation
4210 \eqL@tagging@register@env{latexequation}%
4211 \eqL@provide@moveenv{hyperrefequation}{equation}%
4212 \eqL@tagging@register@env{hyperrefequation}%
4213 }%
4214 }%
4215 \@ifpackageloaded{amsmath}{}{\@ifpackageloaded{hyperref}{}{%
4216 \eqL@provide@moveenv{latexequation}{equation}%
4217 \eqL@tagging@register@env{latexequation}%
4218 }%
4219 }

```

TODO: describe

```

4220 \def\eqL@backup@env@multlined{%
4221 \AddToHook{package/mathtools/after}{%
4222 \eqL@provide@moveenv{amsmultlined}{multlined}%
4223 }%
4224 }

```

TODO: describe

```

4225 \def\eqL@backup@env@subequations{%
4226 \eqL@amsmath@after{%
4227 \eqL@provide@moveenv{amssubequations}{subequations}%

```

```

4228 }%
4229 \AddToHook{package/hyperref/after}{%
4230   \AddToHook{cmd/amssubequations/before}{%
4231     {%
4232       \stepcounter{equation}%
4233       \protected@edef\theHparentequation{\theHequation}%
4234       \addtocounter{equation}{-1}%
4235     }%
4236   \AddToHook{cmd/amssubequations/after}{%
4237     {%
4238       \def\theHequation{\theHparentequation\alph{equation}}%
4239       \ignorespaces
4240     }%
4241   }%
4242 }

```

TODO: describe

```

4243 \def\eql@backup@env{%
4244   \eql@backup@env@equation
4245   \eql@backup@env@amsmath{gather}
4246   \eql@backup@env@amsmath{multline}
4247   \eql@backup@env@amsmath{align}
4248   \eql@backup@env@amsmath{flalign}
4249   \eql@backup@env@amsmath{aligned}
4250   \eql@backup@env@amsmath{gathered}
4251   \eql@backup@env@multlined
4252   \eql@backup@env@subequations
4253 }

```

TODO: describe

```

4254 \def\eql@provide@env@amsmath#1#2{%
4255   \eql@provide@onlyonce{#1}{#2}%
4256   \ifdefined\eql@tmp
4257     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@#1}%
4258   \else
4259     \eql@amsmath@after{%
4260       \eql@provide@moveenv{#1}{\eql@#1}%
4261       \eql@provide@moveenv{#1*}{\eql@#1*}%
4262     }%
4263     \AddToHook{package/mathtools/after}{%
4264       \eql@provide@moveenv{#1}{\eql@#1}%
4265       \eql@provide@moveenv{#1*}{\eql@#1*}%
4266     }%
4267     \eql@provide@moveenv{#1}{\eql@#1}%
4268     \eql@provide@moveenv{#1*}{\eql@#1*}%
4269     \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
4270     \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
4271   \fi
4272 }

```

TODO: describe

```

4273 \def\eql@provide@env@equation#1{%
4274   \eql@provide@onlyonce{equation}{#1}%
4275   \ifdefined\eql@tmp
4276     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@equation}%
4277   \else
4278     \eql@amsmath@after{%
4279       \eql@provide@moveenv{equation}{\eql@equation}%

```

```

4280     \eql@provide@moveenv{equation*}{eql@equation*}%
4281 }%
4282 \AddToHook{package/hyperref/after}{%
4283     \ifpackageloaded{amsmath}{}{%
4284         \eql@provide@moveenv{equation}{eql@equation}%
4285     }%
4286 }%
4287 \eql@provide@moveenv{equation}{eql@equation}%
4288 \eql@provide@moveenv{equation*}{eql@equation*}%
4289 \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
4290 \ifdefined\eql@tagging@on
4291     \AddToHook{begindocument/end}{%
4292         \eql@provide@moveenv{equation}{eql@equation}%
4293         \eql@provide@moveenv{equation*}{eql@equation*}%
4294     }%
4295 \fi
4296 \fi
4297 }

```

TODO: describe

```

4298 \def\eql@provide@env@multlined#1{%
4299     \eql@provide@onlyonce{multlined}{#1}%
4300     \ifdefined\eql@tmp
4301         \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
4302     \else
4303         \AddToHook{package/mathtools/after}{%
4304             \eql@provide@moveenv{multlined}{eql@multlined}%
4305         }%
4306         \eql@provide@moveenv{multlined}{eql@multlined}%
4307         \ifpackageloaded{mathtools}{}{\AddToHook{package/mathtools/before}{%
4308             \eql@provide@undefineenv{multlined}}}%
4309     \fi
4310 }

```

TODO: describe

```

4311 \def\eql@provide@env@subequations#1{%
4312     \eql@provide@onlyonce{subequations}{#1}%
4313     \ifdefined\eql@tmp
4314         \expandafter\eql@provide@moveenv
4315         \expandafter{\eql@tmp}{eql@subequations}%
4316     \else
4317         \eql@amsmath@after{%
4318             \eql@provide@moveenv{subequations}{eql@subequations}%
4319         }%
4320     \AddToHook{package/hyperref/after}{%
4321         \AddToHook{cmd/subequations/before}[hyperref]{}%
4322         \AddToHook{cmd/subequations/after}[hyperref]{}%
4323         \RemoveFromHook{cmd/subequations/before}[hyperref]%
4324         \RemoveFromHook{cmd/subequations/after}[hyperref]%
4325         \AddToHook{begindocument/end}{%
4326             \eql@provide@moveenv{subequations}{eql@subequations}%
4327         }%
4328     }%
4329     \eql@provide@moveenv{subequations}{eql@subequations}%
4330     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
4331 \fi
4332 }

```

TODO: describe

```
4333 \def\eql@provide@sqr{%
4334   \let\[\eql@equations@sqr@open
4335   \let\]\eql@equations@sqr@close
4336   \eql@amsmath@after{%
4337     \let\[\eql@equations@sqr@open
4338     \let\]\eql@equations@sqr@close
4339   }%
4340   \ifdefined\eql@tagging@on
4341     \AddToHook{begindocument/end}{%
4342       \let\[\eql@equations@sqr@open
4343       \let\]\eql@equations@sqr@close
4344     }%
4345   \fi
4346 }
```

TODO: describe

```
4347 \def\eql@provide@ang{%
4348   \let\<\eql@equations@ang@open
4349   \let\>\eql@equations@ang@close
4350 }
```

TODO: describe

```
4351 \eql@define@key{provide}{equation} [] {\eql@provide@env@equation{#1}}
4352 \eql@define@key{provide}{gather} [] {\eql@provide@env@amsmath{gather}{#1}}
4353 \eql@define@key{provide}{multline} [] {\eql@provide@env@amsmath{multline}{#1}}
4354 \eql@define@key{provide}{align} [] {\eql@provide@env@amsmath{align}{#1}}
4355 \eql@define@key{provide}{flalign} [] {\eql@provide@env@amsmath{flalign}{#1}}
4356 \eql@define@key{provide}{aligned} [] {\eql@provide@env@amsmath{aligned}{#1}}
4357 \eql@define@key{provide}{gathered} [] {\eql@provide@env@amsmath{gathered}{#1}}
4358 \eql@define@key{provide}{multlined} [] {\eql@provide@env@multlined{#1}}
4359 \eql@define@key{provide}{subequations} [] {\eql@provide@env@subequations{#1}}
4360 \eql@define@key{provide}{sqr} [] {\eql@provide@sqr}
4361 \eql@define@key{provide}{ang} [] {\eql@provide@ang}
4362 \eql@define@key{provide}{eqref} [] {\eql@provide@cmd{eqref}{#1}}
4363 \eql@define@key{provide}{tagform} [] {%
4364   \def\tagform@##1{\maketag@@@{\eql@tag@form{#1}}}}
4365 \eql@define@key{provide}{maketag} [] {%
4366   \def\maketag@@@##1{\eql@tag@box{##1}}}
```

TODO: describe

```
4367 \newcommand{\eqnlinesprovide}[1]{%
4368 (dev)\eql@dev@start\eqnlinesprovide
4369   \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:

```
4370 \newcommand{\eqnlineset}[1]{%
4371 (dev)\eql@dev@start\eqnlineset
4372   \eql@setkeys{setup}{#1}}
```

Disable error message for exclusive package options:

```
4373 \let\eq1@error@packageoption@gobble
```

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

```
4374 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
4375 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
4376 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
4377 \eq1@defaults@eqnlines
4378 \eq1@mode@columns
4379 \eq1@mode@aligned
```

Process package options:

```
4380 \ProcessOptions
```

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

```
4381 \def\eq1@error@packageoption#1{%
4382   \eq1@error{may only use ‘#1’ as a package option}%
4383 }
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the T_EX conditional parsing mechanism:

```
4384 \ifdefined\tagsleft@true\else
4385   \expandafter\newif\csname iftagsleft@\endcsname
4386 \fi
4387 \ifdefined@fleqntrue\else
4388   \expandafter\newif\csname if@fleqn\endcsname
4389 \fi
```

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

```
4390 \ifdefined\eq1@provide@opt@amsmath
4391   \let\eq1@provide@opt@equation\eq1@true
4392   \eq1@amsmath@after{%
4393     \iftagsleft@
4394       \eqnlineset{tagsleft}
4395     \else
4396       \eqnlineset{tagsright}
4397     \fi
4398     \if@fleqn
4399       \eqnlineset{left}
4400     \else
4401       \eqnlineset{center}
4402     \fi
4403   }
4404 \fi
```

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

```
4405 \ifdefined\eq1@provide@opt@amsmathends\eq1@amsmath@fixends\fi
```

TODO: describe

```
4406 \ifdefined\eq1@provide@opt@backup\eq1@backup@env\fi
```

Provide native L^AT_EX environment `equation` and symbolic shortcut `\[...\]` if desired:

```
4407 \ifdefined\eql@provide@opt@equation\eqnlinesprovide{equation,sqr}\fi
```

Provide `amsmath` equation environments if desired:

```
4408 \ifdefined\eql@provide@opt@amsmath
4409   \eqnlinesprovide{%
4410     multiline,gather,align,flalign,%
4411     multlined,gathered,aligned,%
4412     subequations}
4413 \fi
```

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

```
4414 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
4415 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```