

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`

2025/06/23, v0.10.1

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

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

Abstract

`eqnlines` is a $\text{\LaTeX} 2_{\epsilon}$ package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of \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	8
2.3	Horizontal Placement	12
2.4	Punctuation	17
2.5	Math Classes at Alignment	17
2.6	Vertical Spacing	18
2.7	Further Environments and Features	21
2.8	General Options	24
2.9	Feature Selection and Package Options	24
3	Information	25
3.1	Copyright	25
3.2	Credits	25
3.3	Files and Installation	25
3.4	Related CTAN Packages	26
3.5	Feature Suggestions	27
3.6	Revision History	27
A	Implementation	29

B	General Support	29
B.1	Development Messages	29
B.2	Supporting Definitions	29
B.3	Dollardollar Abstraction	30
B.4	Look-Ahead in Alignment	30
B.5	Error Messages	32
B.6	amsmath Integration	33
B.7	PDF Tagging Support	33
B.8	Key-Value Processing	35
C	Parameters and Registers	37
C.1	Parameters	37
C.2	Registers	40
C.3	Hooks	43
D	Features	43
D.1	Punctuation	43
D.2	Math Classes at Alignment	45
D.3	Framed Cells	45
D.4	Alternative Content Description	46
E	Equation Numbering	46
E.1	Supporting Definitions	46
E.2	Schemes	47
E.3	Interface	51
E.4	Integration	55
E.5	Positioning	58
E.6	Component Display	62
E.7	Tag Composition	65
F	Subequation Numbering	68
F.1	Definitions	68
F.2	Environment	69
F.3	Subequation Scheme	70
G	Display Equations Support	71
G.1	Display Breaks	71
G.2	Explicit Vertical Space	72
G.3	Default Vertical Spacing	74
G.4	Entry and Exit	74
G.5	Stack	80
H	Multi-Line Support	81
H.1	Measure Support	81
H.2	Line Breaks	83
H.3	Intertext	83
H.4	Line Marks	86
I	Column Placement	87
I.1	Supporting Definitions	87
I.2	Shape Schemes	87
I.3	Width Data	91
I.4	Best Line Selection	94
I.5	Tag Margin	95
I.6	Single Column	96

I.7	Multiple Columns	97
J	Single Column Arrangement	103
J.1	Supporting Definitions	103
J.2	Arrangement Methods	103
J.3	Central Alignment	105
J.4	Left Alignment	106
J.5	Right Alignment	107
K	Equations Box Environment	109
K.1	Line Breaks	109
K.2	Stacked Mode	109
K.3	Aligned Mode	110
K.4	Main	112
K.5	Environment	114
L	Single-Line Equation	114
L.1	Native Mode	114
L.2	Print	115
M	Multi-Line with Single Column	116
M.1	Measure	116
M.2	Column Placement	118
M.3	Print	118
N	Multi-Line with Multiple Columns	120
N.1	Support	120
N.2	Transpose	120
N.3	Measure	122
N.4	Columns Placement	125
N.5	Print	125
O	Interface	129
O.1	Scanning the Equation Body	129
O.2	Options Processing	132
O.3	Single-Line Main	133
O.4	Multi-Line Main	135
O.5	Equations Environment	136
P	Options	139
P.1	Selection Tools	139
P.2	Options Declarations	139
P.3	Parameter Presets	150
P.4	Component Selection	152
P.5	Global and Package Options	160

1 Introduction

Typesetting mathematical equations is an undisputed strength of $\text{T}_{\text{E}}\text{X}$. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ improved the overall management of display equations, for instance by providing optional numbering. It also added elementary functionality for multi-line equations with alignment. Some of its deficiencies were addressed by the multi-line equation environments of the package `amsmath` which have become an established standard for these purposes.

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

- Equation numbers can be assigned to individual lines (as for `align` and `gather`) or once for the multi-line equation block (as for `multline`). In the former case, a sub-numbering scheme can be applied (as through `subequations`). In the latter case, the position can be assigned to a specific line (first/middle/last/chosen). Moreover, equation numbers can be turned on and off by commands, and they can be triggered by setting a label.
- The vertical spacing above and below single- and multi-line equations of `LATEX` and `amsmath` can be somewhat variable, hard to control and even resistive in certain situations. The package implements clearer structures controlling the vertical spacing, including proper dependency on the text line above and ways to adjust the spacing.
- The framework introduces a scheme which semi-automatically inserts punctuation, e.g. ‘.’ or ‘,’, at the end of the following (or every) equation environment. Punctuation can also be inserted at every alignment column or equation line including the possibility to prepend a certain spacing.
- Next to `\[...\]` as an alias for the single-line `equation` environment, the package uses `\<...\>` as an alias multi-line equations.
- The horizontal alignment and indentation of equation lines can be adjusted via a scheme or on a line-by-line basis.
- The alignment marker can be placed before or after the equation signs while maintaining proper spacing to symbols before and after it. This simplifies the construction of continuing equations in an aligned context.
- Equation lines are subject to shrinking of space if the available space does not suffice (analogously to single-line equations).
- Most settings can be controlled via optional arguments and modifiers to the equation environment or via global settings. This includes switching between different types of equation environments, enabling or disabling numbering, adjusting vertical spacing, etc. This feature simplifies the adjustment and fine-tuning of equations towards the intended layout.
- Last but not least, the underlying `amsmath` code, originating from the `TEX` era and early `LATEX` years, has been redesigned with emphasis on clarity, readability, adjustability and maintainability (but at the cost of moderately higher resource consumption and moderately lower efficiency). Nevertheless, it remains original `LATEX 2ε` code without using the `expl3` layer.

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

2 Usage

Notice regarding package version v0.10.1: Please note that this package is still in a development and testing stage in the present version. This mainly applies to the documen-

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

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

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

To use the eqnlines package add the command

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

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

2.1 Equations Environment

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

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

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

We note that the `equations` environment should be started with a whitespace character ‘`\`’ which provides a clear separation from optional arguments ‘*[opts]*’ and/or modifiers which must immediately follow the environment declaration `\begin{equations}` without whitespaces. Any character without a proper meaning will also start the equation content, however, future versions of the package may extend the syntax of modifiers, and thus a separation by whitespace is advertised.

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

$$\backslash\text{eqnlineset}\{\textit{opts}\}$$

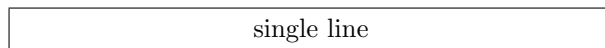
`\eqncontrol` Several options can be controlled for individual lines or cells within the equations block by the macro:

$$\backslash\text{eqncontrol}\{\textit{opts}\}$$

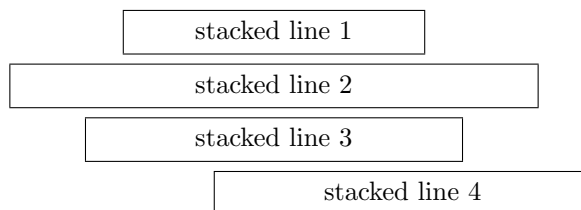
The `\eqncontrol` interface also provides several features for which no other macro definitions exist. Shortcuts to frequently used features could be installed by user definitions such as:

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

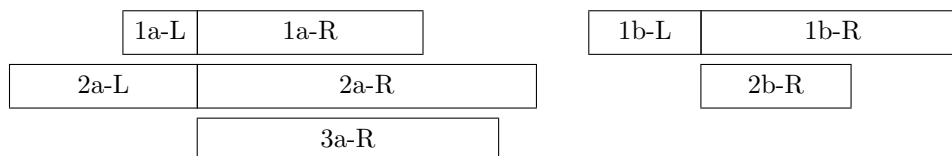
Modes of Operation. The package supplies a main maths environment called `equations` which has three principal modes of operation. It can display a single-line equation just as the L^AT_EX environment `equation` or the symbolic shortcut `\[...\]`:



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



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



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

`lines` (*key*)

`columns` (*key*)

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

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

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

$$x = \cos \phi \tag{1}$$

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.

```

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

```

$$x = \cos \phi \quad (2)$$

$$\phi = \arccos x \quad (3)$$

$$x = \cos \phi \quad \phi = \arccos x \quad (4)$$

$$= (z + z^{-1})/2 \quad = -i \log z \quad (5)$$

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

```

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

```

$$x = \cos \phi$$

$$x = \cos \phi$$

$$\phi = \arccos x$$

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

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

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

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

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

```

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

```

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

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

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

$$x = \cos \phi \quad \phi = \arccos x \quad (10)$$

$$= (z + z^{-1})/2 \quad = -i \log z \quad (11)$$

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

```

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

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

```

2.2 Numbering

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

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

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

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

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

`\nonumber` and `\donumber`

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

`nn,*` (*key*) `\[*_...\]` and `\[!_...\]`
`dn,!` (*key*)

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

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

`\numberhere` and `\numbernext`

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

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

$$x = \cos \phi$$

$$= (z + z^{-1})/2 \tag{27}$$

$$\phi = \arccos x$$

$$= -i \log z \tag{28}$$

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

$$x = \cos \phi \tag{29}$$

```
\phi &= \arccos x \\
  &= -i\log z
\end{equations}
```

$$\phi = \arccos x \tag{30}$$

$$= -i \log z$$

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

$$x = \cos \phi$$

$$\phi = \arccos x \tag{31}$$

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

$$x = \cos \phi$$

$$\phi = \arccos x$$

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

$$x = \cos \phi$$

$$= (z + z^{-1})/2 \tag{32}$$

$$\phi = \arccos x$$

$$= -i \log z \tag{33}$$

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

$$x = \cos \phi$$

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

$$\phi = \arccos x \tag{34}$$

$$= -i \log z$$

```

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

```

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

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

\label[*name*]{*label*} and **\tag**[*][*ref*]{*tag*}

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

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

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

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

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

\@{*label*}...**\@**

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

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

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

\eqref{*label*}.

\tagform For custom typesetting, **\tagform** encloses a number/tag with decoration, **\tagbox** puts the decorated number in a box and **\tagboxed** combines the two.
\tagbox
\tagboxed
tagbox (*key*) The typesetting of equation numbers and tags passes through two macros, one which defines the layout and another one which adds a decoration by parentheses. These two methods
tagform (*key*) can be adjusted via the options:

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

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

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

```

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

```

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

2.3 Horizontal Placement

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

left (key)

```

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

```

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

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

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

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

<code>\eqnlineset{tagmarginthreshold=0.7}</code>		
<code>\[! \framebox[4em]{}</code>		(45)
<code>\[! \framebox[8em]{}</code>		(46)
<code>\[! \framebox[12em]{}</code>		(47)
<code>\[! \framebox[16em]{}</code>		(48)

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

<code>\eqnlineset{layout=left}</code>		
<code>\<</code>	$x = \cos \phi$	
<code> x &= \cos\phi \ \ \</code>	$= (z + z^{-1})/2$	
<code> &= (z+z^{-1})/2 \ \ \</code>	$\phi = \arccos x$	
<code>\phi &= \arccos x \ \ \</code>	$= -i \log z$	
<code> &= -i\log z</code>		
<code>\></code>		
<code>\<[tags=left,!]</code>	(49) $x = \cos \phi$	
<code> x &= \cos\phi \ \ \</code>	(50) $= (z + z^{-1})/2$	
<code> &= (z+z^{-1})/2 \ \ \</code>	(51) $\phi = \arccos x$	
<code>\phi &= \arccos x \ \ \</code>	(52) $= -i \log z$	
<code> &= -i\log z</code>		
<code>\></code>		

fulllength (*key*) **Column Separation.** The horizontal alignment of columns is fixed for aligned multi-line equations: Each pair of subsequent columns forms a unit which is aligned at the intermediate alignment marker '&'. These columns are distributed evenly over the available horizontal space. Here, the outer space left and right of the set of columns is treated on equal footing to the space between the columns (option `fulllength=off`; default), but it can be eliminated so that the outer columns are pushed right to the margin (option `fulllength=on`). A minimum and maximum column separation can be specified via `mincolsep=dimen` and `maxcolsep=dimen` (defaults are `2em` and `1em`) or the maximum column separation can be disabled by `maxcolsep=off` (which is implied by `fulllength=on`).

`\<[maxcolsep=2em]`

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

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

`\<[maxcolsep=off]`

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

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

`\<[fulllength]`

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

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

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

name	alt.	shape	alignment
<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 replacement `amsmath` environment `multline`.

`\eqnlineset{pad=2em}`

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

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

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

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

`shifto` (*key*) `\eqncontrol{align=left|center|right}`
`shiftby` (*key*) `\eqncontrol{shifto|shiftby=dimen}`

`\shoveleft` or by the macros:

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

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

`indent` (*key*) or hanging indentation by the standard indentation amount (`indent=2em`). Furthermore, `\shoveby[*]{dimen}` shifts the line by the additional amount *dimen* (the star variant shifts to an absolute position relative to the reference position).

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

```

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

```

Fitting. Finally, we note that the package will make attempts at fitting the equation components into the horizontal space by adjusting some dimensions with the priority of avoiding overlong lines. The adjustments will first concern the intercolumn and margin

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

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

`\<=!` $x + x$ (53)

`x+x \\\` $x + x + x + x$ (54)

`x+x+x+x \\\` $x + x + x + x + x + x$ (55)

`x+x+x+x+x+x \\\` $x + x + x + x + x + x + x + x$ (56)

`x+x+x+x+x+x+x+x \\\` $x + x + x + x + x + x + x + x + x + x$ (57)

`x+x+x+x+x+x+x+x+x+x \\\` $x + x + x + x + x + x + x + x + x + x + x + x$ (58)

`\>`

`mintagsep` (*key*) If the available space on a line does not suffice to place both the equation and its tag (with a minimum separation of `mintagsep`; default is `0.5em`), a tag will automatically be shifted (lowered or raised depending on whether it is placed on the right or left) to an otherwise empty line. The `\eqncontrol` control `shifftag=dimen` (alternatively `\raisetag*`) may be

`\raisetag*` used to shift a tag up (or down with negative arguments). The control `smashtag=dimen` (alternatively `\raisetag`) may be used to fine-tune the vertical placement when the tag

`\raisetag` requires extra vertical space but some space above or below the tag is unoccupied. It smashes some of the tag's height (or depth with negative arguments) and thus reduces the vertical gap created by the tag. Note that this feature can be used successively with positive

`pushtag` (*key*) and negative arguments to reduce the space in both directions if available. Where needed, the control `pushtag` (or `\raisetag!`) force-pushes the tag to a separate line and frees up the horizontal space occupied by the tag. The numbering modes `top`, `bottom`, `center`, `median`, `center!` and `center*` are special in that they allow for a continuous vertical placement of

`tagbetween` (*key*) the tag between two lines. The more flexible placement of tags may also be enabled for the single-lines modes by the option `tagbetween`. Here, both lines must have sufficiently much space available for the tag. If not, the tag is shifted up or down or it is placed on separate

`tagsnap` (*key*) line between the two. The option `tagsnap` defines a range within which the tag baseline snaps to a nearby math baseline.

`\[! \phi = -\int \frac{dx}{\sqrt{1+x^2}} \]` $\phi = -\int \frac{dx}{\sqrt{1+x^2}}$ (59)

`\[! x = \frac{\partial}{\partial \phi} \sin \phi \]` $x = \frac{\partial}{\partial \phi} \sin \phi$ (60)

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

2.4 Punctuation

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

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

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

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

<pre>\eqnlineset{punct=.</pre> <p>The equation</p> <pre>\[x = \cos\phi \eqnpunct{}</pre> <p>can also be written as</p> <pre>\eqnpunct,</pre> <pre>\[x = (z+z^{-1})/2 \]</pre> <p>where we assume</p> <pre>\[z = \exp(i\phi) \]</pre>	<p>The equation</p> $x = \cos \phi$ <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 “QED”, it can be invoked manually by `\eqnpunctapply`.

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

`punctcol` (*key*) For multi-line equations, there are two further levels of default punctuation for terminating columns and lines which are specified via the macros `\eqnpunctcol` and `\eqnpunctline` or 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*`.

```
\eqnlineset{punct={.},
  punctcol={,},punctline={;}}
\< x &= \cos\phi &
\phi &= \arccos x \\\
  x &= (z+z^{-1})/2 &
\phi &= -i\log z \>
```

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

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

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

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 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	_a-out		_in-b	b		
				_lead-c	c			

2.6 Vertical Spacing

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

The spacing is determined by combination of several aspects:

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

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

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

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

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

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

<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: <code>\[[short] \mbox{forced short} \]</code> and short: <code>\[[long] \mbox{forced long} \]</code> 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.

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

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

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

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

Glue Dimensions. The package also maintains several global vertical space settings

`...skip` (*key*) `aboveposskip` and `belowposskip` (sometimes `posskip` for both):

<code>...posskip</code>	both	description
<code>...long...</code>	<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>	minimum glue for outer raised/lowered tags

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

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

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

2.7 Further Environments and Features

The package supplies some additional environments and features:

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

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

`colsep (key)`

`frame (key)` `\[\left\{`

`wrap (key)` `\begin{equationsbox}[margin=1em]`

`x &= \cos\phi \ \`

`\phi &= \arccos x`

`\end{equationsbox}`

`\right\}\]`

`$$\Longrightarrow\<=[shape=1,frame]`

`x = \cos\phi &`

`\phi = \arccos x \ \`

`x = (z+z^{-1})/2 &`

`\phi = -i\log z`

`\>\Longleftarrow$`

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

\Rightarrow

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

\Leftarrow

`subequations` (*env.*) **Collective Numbering.** The environment `subequations` groups equations contained in the body with a common primary equation number and an extra level of numbering (typically: a, b, c, ...). The numbering layout can be controlled via `subeqtemplate`. For instance, the default behaviour of adding lowercase latin letters to the parent equation number (#1) is achieved by:

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

```

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

```

$$x = \cos \phi \quad (62\text{-i})$$

$$\phi = \arccos x \quad (62\text{-ii})$$

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

```

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

```

$$x = \cos \phi$$

and

$$\phi = \arccos x$$

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

```

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

```

$$x = \cos \phi$$

$$\phi = \arccos x$$

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

$$\backslash\text{eqncontrol}\{\text{markline}=\{\text{symbol}=\text{sym}, \text{opts}\}\}$$

$$\backslash\text{eqncontrol}\{\text{qed}[\text{opts}]\}$$

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

```

\<[n=1]! x &= \cos\phi
\eqncontrol{markline={symbol=\sqrt{\$}} \>
\phi &= \arccos x
\eqncontrol{qed={shift=.5ex}} \>

```

$$x = \cos \phi$$

$$\phi = \arccos x$$

√
(63)
QED

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

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

`\framecell[cmd]` or `\eqncontrol{framecell[=cmd]}`

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

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

`\eqncontrol{frametag[=cmd]}`

<code>\< x &= \cos\phi \\ \framecell \phi &= \arccos x \></code>	$x = \cos \phi$ <div style="border: 1px solid black; padding: 2px; display: inline-block;">$\phi = \arccos x$</div>
<code>\[\framecell[\fboxrule2pt\fbox] \mbox{important} \eqnpunct! \]</code>	<div style="border: 2px solid black; padding: 2px; display: inline-block;">important!</div>
<code>\[! \framecell[\fcolorbox{white}{yellow}] \eqncontrol{frametag=\fboxsep2pt\fbox} \mbox{highlight}\]</code>	<div style="background-color: yellow; padding: 2px; display: inline-block;">highlight</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">(64)</div>

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

`alt={alt text}` or `\eqnalt[opt]{alt}`

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

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

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 value <code>reuse</code> avoids third measuring pass
<code>ampproof</code>	equip optional argument parsing with protection for ‘&’
<code>crerror</code>	invoke an error when ‘\’ is used in a single equation
<code>modifierwarning</code>	invoke a warning for unknown environment modifiers
<code>scanpar</code>	allow scanning of <code>\par</code> within equation body (e.g., for use in nested <code>\parbox</code> or <code>minipage</code>)

2.9 Feature Selection and Package Options

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

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

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

```
\eqnlinesprovide{features}
```

features is a comma-separated list of features:

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

3 Information

3.1 Copyright

Copyright © 2024–2025 Niklas Beisert

Based on the L^AT_EX package `amsmath`: Copyright © 1995, 2000, 2013 American Mathematical Society; 2016–2024 L^AT_EX 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 L^AT_EX 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 emphasising equations via `empheq` do not (yet) work.
- This package cooperates with the package `hyperref` to create anchors and references within the electronic document. Compatibility with the `hyperref` package has been tested with v7.011 (2024/11/05).
- This package cooperates with the package `beamer` in assigning default colours for math content. Compatibility with the `beamer` package has been tested with v3.74 (2025/06/15).
- This package supports the display of labels and references through the package `showkeys`. Compatibility with the `showkeys` package has been tested with v3.21 (2024/05/23).
- This package supports placement of QED symbols within proofs through the `\qedhere` interface of the package `amsthm`. Compatibility with the `amsthm` package has been tested with v2.20.6 (2020/05/29).
- This package is currently not compatible with the package `cleveref` (thanks to Jonáš Dujava for pointing out). The command `\Cref` will not refer properly to equation numbers recorded by the `equations` environment. Further features of either package and/or/in combination with `amsmath` may fail due to the patching by the package. The alternative package `zref-clever` appears to work as intended. Incompatibility with the `cleveref` package has been observed for v0.21.4 (2018/03/27). Compatibility with the `zref-clever` package has been tested with v0.5.1 (2024/11/28).

3.5 Feature Suggestions

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

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

3.6 Revision History

v0.10.1: 2025/06/23

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

v0.10: 2025/05/29

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

v0.9: 2025/05/18

- option `transpose` to transpose rows and columns in columns mode (thanks to Christophe Bal for suggestion)
- added `\eqncontrol` interface for control within lines and cells
- internal structure and interface changes
- added `\vspace*` for persistent glue at page breaks
- added framed tags (`frametag`)
- added `\raisetag!` to enforce raising (or lowering) of tags even if space is sufficient
- added modifiers, relaxed order, changed lines mode modifier from `'~'` to `'='`
- fixed minor issues
- thanks to Jonáš Dujava for various reports and suggestions

v0.8: 2025/04/30

- added framed cells (`\framecell`)
- added automatic best line selection for tag placement (`best` and `evadetag`)
- symbolic environment `\<...>` forwards to `equationsbox` in math mode

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

v0.7.1: 2025/04/09

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

v0.7: 2025/04/03

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

v0.6.1: 2025/03/27

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

v0.6: 2025/03/11

- preliminary PDF tagging support (<https://latex3.github.io/tagging-project/>); `amsmath` *must* be loaded *before* `eqnlines` to avoid errors
- classic L^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
- thanks to Till Bargheer for testing and reports

A Implementation

The appendix documents the various components of the present package.

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

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

B General Support

In the following we describe general purpose supporting routines.

B.1 Development Messages

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

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

B.2 Supporting Definitions

`\eql@false` (*bool*) Rather than the standard L^AT_EX scheme of `\xxxfalse`, `\xxxtrue` and `\ifxxx` for boolean variables *xxx*, we use a scheme where `\xxx` is either undefined or defined (to an empty macro) and is tested against by the ϵ -T_EX conditional `\ifdefined\xxx`. In order to make the scheme more tangible, we define the two expected values for boolean variables:

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

TODO: describe

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

B.3 Dollardollar Abstraction

`\dollar@begin` As of 2025 L^AT_EX defines `\dollar@begin` and `\dollar@end` to represent
`\eql@dollar@end` (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:

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

B.4 Look-Ahead in Alignment

Scanning for optional arguments [...] or modifiers such as `‘*’` using the L^AT_EX
`\@ifnextchar` mechanism has two challenges within aligned equations: a square bracket or
star may well be part of the intended mathematical expression and the look-ahead could
trip upon an alignment character `‘&’` which inadvertently triggers to enter the next
alignment column.

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

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

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

```
34 \let\eql@atxi=@
35 \begingroup
36   \makeatother
37   \let\tmp=@%
38   \makeatletter
39   \global\let\eql@atxii\tmp
40 \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
`\eql@testopt@...` `\@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 [...]:

```

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

```

TODO: describe

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

TODO: describe

```

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

```

`\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@srbggroup` 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@srbggroup` 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>.

```

81 \def\eq@srbgroup{\iffalse{\fi\ifnum0='}\fi}
82 \def\eq@spegroup{\ifnum0='{ \fi\iffalse}\fi}
83 \def\eq@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
84 \def\eq@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}

```

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

```

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

```

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

```

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

```

`\eq@amproof` `\eq@amprevert` 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:

```

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

```

B.5 Error Messages

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

```

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

```

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

```

114 \def\eq@warn@here#1{\eq@warning{\string#1 not allowed outside equations}}
115 \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 116 \def\eq@warn@tags@unused#1#2{\eq@warning{Unused equation #1:
117 #2 will be lost}}
\eq@warn@tag@multiple 118 \def\eq@warn@tags@multiple#1#2#3{\eq@warning{Multiple equation #1:
119 previous #2 will be lost#3}}
\eq@warn@name@unused 120 \def\eq@warn@label@unused{\eq@warn@tags@unused{\string\label}
121 {label '\eq@tags@label'}}
\warn@name@multiple 122 \def\eq@warn@label@multiple#1{\eq@warn@tags@multiple{\string\label's}
123 {label '\eq@tags@label'}}{ and replaced by '#1'}}
\eq@warn@ref@unused 124 \def\eq@warn@name@unused{\eq@warn@tags@unused{label name}
125 {name declaration}}
\eq@warn@ref@multiple 126 \def\eq@warn@name@multiple{\eq@warn@tags@multiple{label names}
127 {name declaration}}{}
128 \def\eq@warn@tag@unused{\eq@warn@tags@unused{\string>tag}
129 {tag declaration}}
130 \def\eq@warn@tag@multiple{\eq@warn@tags@multiple{\string>tag's}
131 {tag declaration will be lost}}{}
132 \def\eq@warn@ref@unused{\eq@warn@tags@unused{tag label}
133 {tag label declaration}}
134 \def\eq@warn@ref@multiple{\eq@warn@tags@multiple{tag labels}
135 {tag label declaration}}{}

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

```

B.6 amsmath Integration

`\eq@amsmath@after` We need to overwrite certain macros from amsmath. The method `\eq@amsmath@after`

`\eq@amsmath@before` executes argument #1 after loading amsmath is loaded. It also runs the code if amsmath

`\eq@amsmath@undefine` has already been loaded. Furthermore, loading amsmath requires certain macros to be

`\eq@amsmath@let` undefined. To this end `\eq@amsmath@before` will execute argument #1 before any future

loading of amsmath. `\eq@amsmath@undefine` undefines a macro in this way and

`\eq@amsmath@let` overwrites a macro of `\amsmath/`:

```

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

```

TODO: temporary fix for development stages

```

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

```

B.7 PDF Tagging Support

`\eq@tagging@...` Proper PDF tagging⁴ support requires a L^AT_EX (development) version at least of 2025. For

the time being, we define an abstraction layer so that the package will collaborate with

L^AT_EX versions around 2020: **TODO:** adjust to further developments

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

B.8 Key-Value Processing

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

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

Value Selection.

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

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

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

```

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

```

```

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

```

Decide between `true` and `false` or related pairs of values:

```

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

```

`\eql@decide@if`

```

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

```

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

```

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

```

Key Declaration.

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

```

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

```

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

```

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

```

Options and Control Interface.

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

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

`\eqnaddopt`

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

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

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

`\eql@control@default`

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

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

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

C Parameters and Registers

In the following, we collect parameter and register definitions.

C.1 Parameters

TODO: describe

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\eql@skip@force@below`

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

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. \TeX '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.

```

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

```

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

```

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

```

Multi-Line Align Mode.

```

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

```

C.2 Registers

TODO: describe

General. **TODO:** describe

```

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

```

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

TODO: describe

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

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

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

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

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

```
379 \newdimen\eql@line@height@
380 \newdimen\eql@line@depth@
```

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

Rows and Columns.

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

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

```
\eql@column@
\eql@totalcolumns@ 393 \newcount\eql@column@
394 \newcount\eql@totalcolumns@
```

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

```
395 \newdimen\eql@colsep@
```

`\intercolumns@` (*counter*)

```
396 \newcount\eql@intercolumns@
```

Vertical Spacing Adjustments.

`\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:

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

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

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

Shared Registers.

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

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

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

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

C.3 Hooks

`\eql@hook@...` For what it's worth, we define a couple of entry points where one might hook into the equations typesetting framework. The \LaTeX hook framework would be more versatile, but as the purpose of these hooks is rather unclear at the moment, we make this as efficient as it could get: **TODO:** may add a few more hooks

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

D Features

D.1 Punctuation

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

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

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

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

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

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

```
436 \def\eqnpunct{%
437   \eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
438 \def\eql@punct@next@set#1{%
439   \ifmode
440     \def\eql@punct@col{#1}%
441     \def\eql@punct@line{#1}%
442     \def\eql@punct@block{#1}%
443   \else
444     \eqnadopt{punct={#1}}%
445   \fi
446   \ignorespaces}
```

```

447 \def\eql@punct@next@setrelax{%
448   \ifmode
449     \let\eql@punct@block\relax
450   \else
451     \eqnaddopt{punct*}%
452   \fi
453   \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:

```

454 \def\eql@punct@apply@col{%
455   \ifx\eql@punct@col\@empty\else
456     \eql@punct@sep
457     \eql@punct@col
458     \let\eql@punct@col\@empty
459   \fi
460 }

```

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`

```

461 \def\eql@punct@apply@line{%
462   \ifx\eql@punct@line\relax
463   % \TODO hand down immediately?
464   \else
465     \ifx\eql@punct@line\@empty\else
466       \eql@punct@sep
467       \eql@punct@line
468     \fi
469     \let\eql@punct@line\relax
470     \let\eql@punct@col\@empty
471   \fi
472 }

```

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

```

473 \def\eql@punct@apply@block{%
474   \ifx\eql@punct@block\relax
475   % \TODO hand down immediately?
476   \else
477     \ifx\eql@punct@block\@empty\else
478       \eql@punct@sep
479       \eql@punct@block
480     \fi
481     \let\eql@punct@block\relax
482     \let\eql@punct@line\relax
483     \let\eql@punct@col\@empty
484   \fi
485 }

486 \let\eqn@punct@apply\eql@punct@apply@block

```

D.2 Math Classes at Alignment

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

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

```
487 \def\eql@class@innerright@sel@{%
488   \ifdim\eql@prevwidth@=\z@
489     \eql@class@innerlead
490   \else
491     \eql@class@innerright
492   \fi
493 }
```

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

`\class@innerlead@set`

```
494 \def\eql@class@innerleft@set#1{%
495   \def\eql@class@innerleft#{#1}%
496 }
497 \def\eql@class@innerright@set#1{%
498   \def\eql@class@innerright#{#1}%
499   \let\eql@class@innerright@sel\eql@class@innerright
500 }
501 \def\eql@class@innerlead@set#1{%
502   \def\eql@class@innerlead#{#1}%
503   \let\eql@class@innerright@sel\eql@class@innerright@sel@
504 }
```

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

```
505 \def\eql@class@ampeq{%
506   \eql@class@innerleft@set{}%
507   \eql@class@innerright@set{}}%
508 }
509 \def\eql@class@eqamp{%
510   \eql@class@innerleft@set{\mathrel{}}%
511   \eql@class@innerright@set{\mathrel{}}%
512   \eql@class@innerlead@set{}}%
513 }
514 \eql@class@ampeq
```

D.3 Framed Cells

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

```
515 \let\eql@frame@cmd@\undefined
516 \newdimen\eql@frame@margin@
517 \def\eql@frame@set[#1]{%
518   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
519   \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbbox}
520   \def\eql@frame@measure{%
521     \setbox\z@\hbox{\eql@frame@cmd}}%
522   \eql@frame@margin@.5\wd\z@
523 }
524 \def\eql@frame@print{%
```

```

525 \setbox\eql@cellbox@\hbox{%
526   \eql@frame@cmd{\unhbox\eql@cellbox@}%
527 }%
528 }
529 \def\eql@frame@adjust{%
530   \setbox\eql@cellbox@\hbox{%
531     \eql@frame@cmd{%
532       \unhbox\eql@cellbox@
533       \unkern
534       \unskip
535     }%
536     \hfil
537     \kern\z@
538   }%
539 }

```

D.4 Alternative Content Description

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

```

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

```

E Equation Numbering

TODO: describe

E.1 Supporting Definitions

Parameters.

```

541 \let\eql@tags@autolabel\eql@false
542 \let\eql@tags@autotag\eql@true
543 \let\eql@tags@warn\eql@true

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

545 \let\eql@tagpos@doconvert\eql@false

546 \def\eql@tagpos@snap{4pt}

```

Registers.

```

547 \let\eql@numbering@mode\@undefined

548 \let\eql@numbering@active\eql@true
549 \let\eql@numbering@multi\eql@true

550 \let\eql@tags@container\@undefined
551 \def\eql@tags@container@clear{%
552   \let\eql@tags@label\@undefined
553   \let\eql@tags@name\@undefined
554   \let\eql@tags@tag\@undefined
555   \let\eql@tags@ref\@undefined
556   \let\eql@tags@anchor\@empty
557   \eql@tagpos@shift\z@

```

```

558 \eql@tagpos@smashup@z@
559 \eql@tagpos@smashdown@z@
560 \let\eql@tagpos@reserve\eql>true
561 }

562 \let\eql@tags@label@undefined
563 \let\eql@tags@name@undefined
564 \let\eql@tags@tag@undefined
565 \let\eql@tags@ref@undefined
566 \let\eql@tags@frame@cmd@firstofone

```

tags@glabel@ (*counter*)

```

567 \newcount\eql@tags@glabel@
568 \eql@tags@glabel@z@
569 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
570 \def\eql@tags@glabel@step{\global\advance\eql@tags@glabel@\ne}

571 \let\eql@tagpos@continuous\eql>false

572 \newcount\eql@tagpos@row@
573 \newcount\eql@tagpos@prevrow@
574 \newdimen\eql@tagpos@shift@
575 \newdimen\eql@tagpos@smashup@
576 \newdimen\eql@tagpos@smashdown@
577 \newdimen\eql@tagpos@current@
578 \newdimen\eql@tagpos@plain@
579 \newdimen\eql@tagpos@raised@
580 \newdimen\eql@tagpos@target@
581 \newdimen\eql@tagpos@headroom@
582 \newdimen\eql@tagpos@footroom@

```

E.2 Schemes

TODO: describe

Table.

```

583 \def\eql@numbering@tab@sub{sub}
584 \def\eql@numbering@tab@all{all}
585 \def\eql@numbering@tab@first{first}
586 \def\eql@numbering@tab@last{last}
587 \def\eql@numbering@tab@in{in}
588 \def\eql@numbering@tab@out{out}
589 \def\eql@numbering@tab@middle{middle}
590 \def\eql@numbering@tab@best{best}
591 \def\eql@numbering@tab@here{here}
592 \def\eql@numbering@tab@top{top}
593 \def\eql@numbering@tab@bottom{bottom}
594 \def\eql@numbering@tab@center{center}
595 \def\eql@numbering@tab@centerone{centerone}
596 \def\eql@numbering@tab@median{median}
597 \def\eql@numbering@tab@baseline{baseline}

598 \let\eql@numbering@mode\eql@numbering@tab@all
599 \let\eql@numbering@mode@multi\eql@numbering@tab@all
600 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```
601 \let\eqL@numbering@tab@subeq\eqL@numbering@tab@sub
602 \let\eqL@numbering@tab@subequation\eqL@numbering@tab@sub
603 \let\eqL@numbering@tab@subequations\eqL@numbering@tab@sub
604 \let\eqL@numbering@tab@mid\eqL@numbering@tab@middle
605 \let\eqL@numbering@tab@outside\eqL@numbering@tab@out
606 \let\eqL@numbering@tab@inside\eqL@numbering@tab@in
607 \let\eqL@numbering@tab@within\eqL@numbering@tab@in
608 \let\eqL@numbering@tab@opt\eqL@numbering@tab@best
609 \let\eqL@numbering@tab@optimal\eqL@numbering@tab@best
610 \let\eqL@numbering@tab@pick\eqL@numbering@tab@here
611 \let\eqL@numbering@tab@med\eqL@numbering@tab@median
612 \eqL@letcs{eqL@numbering@tab@center*}\eqL@numbering@tab@baseline
613 \eqL@letcs{eqL@numbering@tab@center!}\eqL@numbering@tab@centerone
```

TODO: describe

```
614 \let\eqL@numbering@tab@a\eqL@numbering@tab@all
615 \let\eqL@numbering@tab@s\eqL@numbering@tab@sub
616 \let\eqL@numbering@tab@f\eqL@numbering@tab@first
617 \let\eqL@numbering@tab@l\eqL@numbering@tab@last
618 \let\eqL@numbering@tab@o\eqL@numbering@tab@out
619 \let\eqL@numbering@tab@i\eqL@numbering@tab@in
620 \let\eqL@numbering@tab@h\eqL@numbering@tab@here
621 \let\eqL@numbering@tab@t\eqL@numbering@tab@top
622 \let\eqL@numbering@tab@b\eqL@numbering@tab@bottom
623 \let\eqL@numbering@tab@c\eqL@numbering@tab@center
624 \let\eqL@numbering@tab@m\eqL@numbering@tab@median
625 \eqL@letcs{eqL@numbering@tab@+}\eqL@numbering@tab@best
626 \eqL@letcs{eqL@numbering@tab@m*}\eqL@numbering@tab@middle
627 \eqL@letcs{eqL@numbering@tab@c*}\eqL@numbering@tab@baseline
628 \eqL@letcs{eqL@numbering@tab@c!}\eqL@numbering@tab@centerone
```

Implementations. **TODO:** describe

```
629 \def\eqL@numbering@init@all{\let\eqL@numbering@multi\eqL@true}
```

TODO: describe

```
630 \def\eqL@numbering@init@sub{%
631   \let\eqL@numbering@multi\eqL@true
632   \ifdefined\eqL@subequations@active
633     \let\eqL@numbering@mode\eqL@numbering@tab@all
634   \else
635     \let\eqL@numbering@subeq@use\eqL@true
636   \fi
637 }

638 \def\eqL@numbering@init@first{\eqL@tagpos@row@one}
639 \def\eqL@numbering@init@last{\eqL@tagpos@row@MM}
640 \def\eqL@numbering@init@here{\eqL@tagpos@row@mone}
```

TODO: describe

```
641 \def\eqL@numbering@init@in{%
642   \ifdefined\eqL@tagsleft
643     \eqL@numbering@init@last
644   \else
645     \eqL@numbering@init@first
```



```
646 \fi
647 }
```

TODO: describe

```
648 \def\eq@numbering@init@out{%
649 \ifdefined\eq@tagsleft
650 \eq@numbering@init@first
651 \else
652 \eq@numbering@init@last
653 \fi
654 }
```

TODO: describe

```
655 \def\eq@tagpos@eval@middle{%
656 \ifnum\eq@tagpos@row@=\z@
657 \eq@tagpos@row@numexpr(\eq@totalrows@
658 +\ifdefined\eq@tagsleft\z@\else\@ne\fi)/\tw@\relax
659 \fi
660 }
```

TODO: describe

```
661 \def\eq@tagpos@eval@best{%
662 \ifnum\eq@tagpos@row@=\z@
663 \let\eq@numbering@best@use\eq@true
664 \eq@numbering@init@out
665 \fi
666 }
```

TODO: describe

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

TODO: describe

```
668 \let\eq@numbering@init@top\eq@numbering@init@continuous
669 \def\eq@tagpos@eval@top{%
670 \eq@tagpos@current@\z@
671 }
```

TODO: describe

```
672 \let\eq@numbering@init@bottom\eq@numbering@init@continuous
673 \def\eq@tagpos@eval@bottom{%
674 \eq@tagpos@current@\dimexpr\eq@totalheight@
675 -\eq@tagheight@block@-\eq@tagdepth@block@\relax
676 }
```

TODO: describe

```
677 \let\eq@numbering@init@center\eq@numbering@init@continuous
678 \def\eq@tagpos@eval@center{%
679 \ifnum\eq@totalrows@=\@ne
680 \eq@tagpos@row@\@ne
681 \fi
682 \eq@tagpos@current@\dimexpr(\eq@totalheight@
683 -\eq@tagheight@block@-\eq@tagdepth@block@)/\tw@\relax
684 }
```

TODO: describe

```

685 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
686 \def\eql@tagpos@eval@centerone{%
687   \eql@tagpos@current@\dimexpr(\eql@totalheight@
688     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
689 }

```

TODO: describe

```

690 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
691 \def\eql@tagpos@eval@baseline{%
692   \eql@tagpos@current@\dimexpr(\eql@totalheight@
693     +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
694 }

```

TODO: describe

```

695 \let\eql@numbering@init@median\eql@numbering@init@continuous
696 \def\eql@tagpos@eval@median{%
697   \ifnum\eql@tagpos@row@=\z@
698     \ifodd\eql@totalrows@
699       \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
700     \else
701       \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
702       \eql@dimensions@get\eql@tagpos@row@
703       \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
704         +(\eql@line@interline@-\eql@tagheight@block@
705         +\eql@tagdepth@block@)/\tw@\relax
706     \fi
707   \ifnum\eql@totalrows@=\@ne
708     \eql@tagpos@row@\@ne
709   \else
710     \eql@tagpos@adjust@eval@convert
711     \eql@tagpos@row@\z@
712   \fi
713 \fi
714 }

```

Selection.

```

715 \def\eql@numbering@set#1{%
716   \ifcsname eql@numbering@tab@#1\endcsname
717     \expandafter\let\expandafter\eql@numbering@mode
718     \csname eql@numbering@tab@#1\endcsname
719     \ifx\eql@numbering@mode\eql@numbering@tab@all
720       \let\eql@numbering@mode@multi\eql@numbering@mode
721     \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
722       \let\eql@numbering@mode@multi\eql@numbering@mode
723     \else
724       \let\eql@numbering@mode@single\eql@numbering@mode
725     \fi\fi
726   \else
727     \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
728     \let\eql@numbering@mode\eql@numbering@tab@all
729   \fi
730 }

```

TODO: describe

```

731 \def\eql@numbering@init{%
732   \let\eql@numbering@multi\eql@false

```

```

733 \let\eql@tagpos@continuous\eql@false
734 \let\eql@numbering@subeq@use\eql@false
735 \let\eql@numbering@best@use\eql@false
736 \eql@tagpos@row@\z@
737 \csname eql@numbering@init@\eql@numbering@mode\endcsname
738 \ifdefined\eql@numbering@active
739   \let\eql@numbering@eqnswinit\@eqnswtrue
740 \else
741   \let\eql@numbering@eqnswinit\@eqnswfalse
742 \fi
743 \let\eql@numbering@active\eql@false
744 }

```

E.3 Interface

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

```

745 \eql@amsmath@after{
746   \let\eql@print@eqnum@default\print@eqnum
747   \let\eql@incr@eqnum@default\incr@eqnum
748 }

```

TODO: describe

```

749 \protected\def\donumber{%
750   \if@eqnsw\else
751     \global\@eqnswtrue
752     \ifx\print@eqn\@empty
753       \global\let\print@eqn\eql@print@eqnum@default
754     \fi
755     \ifx\incr@eqn\@empty
756       \global\let\incr@eqn\eql@incr@eqnum@default
757     \fi
758   \fi
759 }

```

TODO: reconsider operation

`\numberhere`

```

760 \protected\def\eql@numberhere{%
761   \ifdefined\eql@numbering@multi
762     \global\@eqnswtrue
763   \else
764     \global\eql@tagpos@row@\eql@row@
765   \fi
766 }

```

TODO: describe

`\numbernext`

```

767 \protected\def\eql@numbernext{%
768   \ifdefined\eql@numbering@multi
769     \global\@eqnswfalse
770   \else
771     \ifnum\eql@tagpos@row@=\eql@row@
772       \global\advance\eql@tagpos@row@\@ne
773     \fi

```

```
774 \fi
775 }
```

Activation Trigger.

```
776 \def\eql@tags@autoenable{%
777   \global\@eqnswtrue
778   \ifnum\eql@tagpos@row@=\m@ne
779     \numberhere
780   \fi
781 }
```

Labels. **TODO:** describe

\eql@label@org

```
782 \let\eql@label@org\label
```

TODO: describe

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

TODO: describe

```
784 \protected\def\eql@label{%
785   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
786 }
```

TODO: describe

```
787 \def\eql@tags@add@labelname[#1]#2{%
788   \def\eql@tmp{#1}%
789   \ifx\eql@tmp\eql@testopt@default\else
790     \eql@tags@add@name{#1}%
791   \fi
792   \eql@tags@add@label{#2}%
793 }
```

TODO: describe

```
794 \def\eql@tags@set@label#1{%
795   \ifdefined\eql@tags@warn
796     \ifdefined\eql@tags@label
797       \eql@warn@label@multiple{#1}%
798     \fi
799   \fi
800   \def\eql@tags@label{#1}%
801 }
```

TODO: describe

```
802 \def\eql@tags@set@name#1{%
803   \ifdefined\eql@tags@warn
804     \ifdefined\eql@tags@name
805       \eql@warn@name@multiple
806     \fi
807   \fi
808   \def\eql@tags@name{#1}%
809 }
```

TODO: describe

```
810 \def\eql@tags@add@label#1{%
811   \ifdefined\eql@tags@autolabel
812     \eql@tags@autoenable
813   \fi
814   \global\eql@appendexpand\eql@tags@container{%
815     \noexpand\eql@tags@set@label{#1}}%
816 }
```

TODO: describe

```
817 \def\eql@tags@add@name#1{%
818   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
819   \global\eql@appendmacro\eql@tags@container\eql@tmp
820 }
```

TODO: describe

```
821 \def\eql@tags@addblock@label#1{%
822   \eql@appendexpand\eql@tags@container@block{%
823     \noexpand\eql@tags@set@label{#1}}%
824 }
```

TODO: describe

```
825 \def\eql@tags@addblock@name#1{%
826   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
827   \eql@appendmacro\eql@tags@container@block\eql@tmp
828 }
```

Tags. **TODO:** describe

`\eql@tag@default`

```
829 \protected\def\eql@tag@default{%
830   \eql@warn@here\tag
831   \eql@tag@gobble
832 }
833 \let\tag\eql@tag@default
```

`\eql@tag@gobble`

```
834 \def\eql@tag@gobble{%
835   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}
```

TODO: describe

```
836 \protected\def\eql@tag{%
837   \eql@ampprotecttwo\eql@teststaropt@tight
838   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
839   \eql@testopt@default
840 }
```

`\eql@tags@add@tagref`

```
841 \def\eql@tags@add@tagref[#1]#2{%
842   \def\eql@tmp{#1}%
843   \ifx\eql@tmp\eql@testopt@default\else
844     \eql@tags@add@ref{#1}%
845   \fi
```

```
846 \eql@tags@add@tag{#2}%
847 }
```

TODO: describe

```
848 \def\eql@tags@set@tag#1{%
849 \ifdefined\eql@tags@warn
850 \ifdefined\eql@tags@tag
851 \eql@warn@tag@multiple
852 \fi
853 \fi
854 \def\eql@tags@tag{#1}%
855 }
```

TODO: describe

```
856 \def\eql@tags@set@ref#1{%
857 \ifdefined\eql@tags@warn
858 \ifdefined\eql@tags@ref
859 \eql@warn@ref@multiple
860 \fi
861 \fi
862 \def\eql@tags@ref{#1}%
863 }
```

TODO: describe

```
864 \def\eql@tags@add@tag#1{%
865 \ifdefined\eql@tags@autotag
866 \eql@tags@autoenable
867 \fi
868 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
869 \global\eql@appendmacro\eql@tags@container\eql@tmp
870 }
```

TODO: describe

```
871 \def\eql@tags@add@ref#1{%
872 \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
873 \global\eql@appendmacro\eql@tags@container\eql@tmp
874 }
```

tags@add@tagform@off

```
875 \def\eql@tags@add@tagform@off{%
876 \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
877 }
```

TODO: describe

```
878 \def\eql@tags@addblock@tag#1{%
879 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
880 \eql@appendmacro\eql@tags@container@block\eql@tmp
881 }
```

TODO: describe

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

TODO: describe

```
886 \def\eq\@tags@addblock@tagform@off{%
887 \eq\@append\eq\@tags@container@block{\let\eq\@tags@tagform\@firstofone}%
888 }
```

Raise Tags.

\raisetag

```
889 \def\eq\@raisetag@default{%
890 \eq\@warn@here\raisetag
891 \eq\@raisetag@gobble
892 }

893 \def\eq\@raisetag@gobble{%
894 \eq\@ampprotecttwo\eq\@ifstar@tight\@gobble\@gobble
895 }
```

TODO: describe

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

897 \def\eq\@raisetag{%
898 \eq\@ampprotecttwo\eq\@ifstar@tight\eq\@tags@add@raiseshift\eq\@raisetag@test
899 }

900 \def\eq\@raisetag@test#1{%
901 \def\@tempa{#1}%
902 \def\@tempb{!}%
903 \ifx\@tempa\@tempb
904 \eq\@tags@add@forceraise
905 \else
906 \eq\@tags@add@raisesmash{#1}%
907 \fi
908 }

909 \def\eq\@tags@add@raiseshift#1{%
910 \global\eq\@appendexpand\eq\@tags@container{%
911 \advance\eq\@tagpos@shift@the\glueexpr#1\relax\relax}%
912 }

913 \def\eq\@tags@add@raisesmash#1{%
914 \dimen@glueexpr#1\relax
915 \ifdim\dimen@<\z@
916 \global\eq\@appendexpand\eq\@tags@container{%
917 \advance\eq\@tagpos@smashdown@the\dimen@\relax}%
918 \else
919 \global\eq\@appendexpand\eq\@tags@container{%
920 \advance\eq\@tagpos@smashup@the\dimen@\relax}%
921 \fi
922 }

923 \def\eq\@tags@add@forceraise{%
924 \global\eq\@append\eq\@tags@container{\let\eq\@tagpos@reserve\eq\@false}%
925 }
```

E.4 Integration

TODO: describe

Support. **TODO:** describe

```
926 \def\eq@numbering@settools{%
927   \let\label\eq@label
928   \let\tag\eq@tag
929   \let\raisetag\eq@raisetag
930   \let\numberhere\eq@numberhere
931   \let\numbernext\eq@numbernext
932 }
```

TODO: not necessary anymore

```
933 \def\eq@numbering@settools@gobble{%
934   \let\label\eq@label@gobble
935   \let\tag\eq@tag@gobble
936   \let\raisetag\eq@raisetag@gobble
937   \let\numberhere\relax
938   \let\numbernext\relax
939 }
```

```
940 \def\eq@numbering@autoblock{%
941   \begingroup
942     \let\eq@tags@warn\eq@false
943     \eq@tags@container@block
944     \ifdefined\eq@tags@autolabel
945       \ifdefined\eq@tags@label
946         \global\@eqnswtrue
947       \fi
948     \fi
949     \ifdefined\eq@tags@autotag
950       \ifdefined\eq@tags@tag
951         \global\@eqnswtrue
952       \fi
953     \fi
954   \endgroup
955 }
```

```
956 \def\eq@numbering@warnunused{%
957   \ifdefined\eq@tags@label
958     \eq@warn@label@unused
959   \fi
960   \ifdefined\eq@tags@name
961     \eq@warn@name@unused
962   \fi
963   \ifdefined\eq@tags@tag
964     \eq@warn@tag@unused
965   \fi
966   \ifdefined\eq@tags@erf
967     \eq@warn@ref@unused
968   \fi
969 }
```

Single Line. **TODO:** describe

```
970 \def\eq@numbering@single@init{%
971   \let\eq@numbering@multi\eq@false
972   \eq@numbering@settools
973   \eq@numbering@eqnswinit
974   \eq@numbering@autoblock
```



```

975 \global\let\eql@tags@container\eql@tags@container@block
976 \let\eql@tags@warn\eql@true
977 }

978 \def\eql@numbering@single@eval{%
979 \ifnum\eql@tagpos@row@=\m@ne
980 \eqnswfalse
981 \fi
982 }

```

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

```

983 \def\eql@numbering@measure@init{%
984 \eql@numbering@settools
985 \ifdefined\eql@numbering@multi\else
986 \eql@numbering@eqnswinit
987 \eql@numbering@autoblock
988 \fi
989 \global\let\eql@tags@container\eql@tags@container@block
990 \let\eql@tags@warn\eql@true
991 }

```

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

```

992 \def\eql@numbering@measure@line@begin{%
993 \ifdefined\eql@numbering@multi
994 \global\eql@numbering@eqnswinit
995 \fi
996 }

```

TODO: describe

```

997 \def\eql@numbering@measure@blocktag{%
998 \ifdefined\eql@numbering@multi
999 \eqnswfalse
1000 \else
1001 \ifnum\eql@tagpos@row@=\m@ne
1002 \eqnswfalse
1003 \fi
1004 \ifnum\eql@totalrows@=\z@
1005 \eqnswfalse
1006 \fi
1007 \fi
1008 }

```

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

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

```

1009 \def\eql@numbering@print@init{%
1010 \let\eql@tags@warn\eql@false
1011 \ifdefined\eql@numbering@multi
1012 \eql@numbering@settools
1013 \global\let\eql@tags@container\eql@tags@container@block
1014 \else
1015 \let\eql@tags@container@block\eql@tags@container
1016 \eql@numbering@settools@gobble
1017 \fi
1018 }

```

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

```
1019 \def\eql@numbering@print@block@begin{%
1020   \ifdefined\eql@numbering@multi\else
1021     \ifnum\eql@tagpos@row@>\z@
1022       \eql@tags@makeblockanchor
1023       \global\eql@appendexpand\eql@tags@container@block{%
1024         \def\noexpand\eql@tags@anchor{%
1025           \unexpanded\expandafter{\eql@tags@anchor}}}%
1026     \fi
1027   \fi
1028   \ifdefined\eql@numbering@subeq@use
1029     \eql@tags@printsubeqlabel
1030   \fi
1031 }
```

TODO: describe

```
1032 \def\eql@numbering@print@line@begin{%
1033   \ifdefined\eql@numbering@multi
1034     \global\eql@numbering@eqnswinit
1035   \fi
1036 }
```

TODO: describe

```
1037 \def\eql@numbering@print@line@eval{%
1038   \ifdefined\eql@numbering@multi
1039     \if@eqnsw
1040       \eql@tags@container
1041     \fi
1042   \else
1043     \ifnum\eql@tagpos@row@=\eql@row@
1044       \@eqnswtrue
1045       \eql@tags@container@block
1046     \else
1047       \@eqnswfalse
1048     \fi
1049   \fi
1050 }
```

E.5 Positioning

TODO: describe

```
1051 \def\eql@tagpos@single@eval{%
1052   \if@eqnsw
1053     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1054     \ifnum\eql@tagpos@row@>\@ne
1055       \eql@tagpos@row@\@ne
1056     \fi
1057     \ifdefined\eql@tagpos@doconvert
1058       \let\eql@tagpos@continuous\eql@true
1059     \fi
1060     \ifdefined\eql@tagpos@continuous
1061       \eql@tagpos@single@eval@continuous
1062     \fi
1063   \else
1064     \eql@tagpos@row@\z@
```

```

1065 \fi
1066 \eql@tagpos@prevrow@z@
1067 \eql@tagpos@headroom@z@
1068 \eql@tagpos@footroom@z@
1069 }

```

TODO: describe

```

1070 \def\eql@tagpos@single@eval@continuous{%
1071 \ifnum\eql@tagpos@row@>z@
1072 \eql@tagpos@target@\eql@tagpos@shift@
1073 \else
1074 \eql@tagpos@target@\dimexpr\eql@line@height@
1075 -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1076 \fi
1077 \eql@tagpos@row@one
1078 \ifdim\ifdim\eql@tagpos@target@<z@-\fi
1079 \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1080 \eql@tagpos@target@z@
1081 \fi
1082 }

```

TODO: describe

```

1083 \def\eql@tagpos@adjust@eval{%
1084 \if@eqnsw
1085 \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1086 \ifnum\eql@tagpos@row@>\eql@totalrows@
1087 \eql@tagpos@row@\eql@totalrows@
1088 \fi
1089 \ifdefined\eql@tagpos@doconvert
1090 \let\eql@tagpos@continuous\eql@true
1091 \fi
1092 \ifdefined\eql@tagpos@continuous
1093 \ifnum\eql@tagpos@row@>z@
1094 \eql@tagpos@adjust@eval@convert
1095 \fi
1096 \eql@tagpos@adjust@eval@continuous
1097 \fi
1098 \else
1099 \eql@tagpos@row@z@
1100 \eql@tagpos@prevrow@z@
1101 \fi
1102 }

```

TODO: describe

```

1103 \def\eql@tagpos@adjust@eval@convert{%
1104 \eql@tagpos@current@z@
1105 \eql@dimensions@for{%
1106 \ifnum\eql@row@<\eql@tagpos@row@
1107 \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1108 +\eql@line@height@+\eql@line@depth@\relax
1109 \fi
1110 \ifnum\eql@row@=\eql@tagpos@row@
1111 \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1112 +\eql@line@height@-\eql@tagheight@block@\relax
1113 \fi
1114 }%
1115 }

```

TODO: describe

```
1116 \def\eq@tagpos@adjust@eval@continuous{%
1117   \dimen@\dimexpr\eq@tagpos@current@-\eq@tagpos@shift@\relax
1118   \eq@tagpos@row@\eq@totalrows@
1119   \eq@tagpos@prevrow@\z@
1120   \eq@tagpos@headroom@\z@
1121   \eq@tagpos@footroom@\z@
1122   \eq@dimensions@for{%
1123     \ifnum\eq@tagpos@row@=\eq@totalrows@
1124       \eq@tagpos@headroom@\eq@line@interline@
1125       \eq@tagpos@target@\dimexpr\eq@line@interline@
1126         +\eq@line@height@-\dimen@-\eq@tagheight@block@\relax
1127       \ifdim\ifdim\eq@tagpos@target@<\z@-\fi
1128         \eq@tagpos@target@<\glueexpr\eq@tagpos@snap@\relax
1129         \advance\dimen@\eq@tagpos@target@
1130         \eq@tagpos@target@\z@
1131       \fi
1132     \ifdim\eq@tagpos@target@>%
1133       \ifdefined\eq@tagleft-1sp@\relax\else\z@\fi
1134       \eq@tagpos@row@\eq@row@
1135       \eq@tagpos@prevrow@\numexpr\eq@row@-\@ne@\relax
1136     \fi
1137     \advance\dimen@-\dimexpr\eq@line@interline@
1138       +\eq@line@depth@+\eq@line@height@\relax
1139   \fi
1140   \ifnum\eq@row@=\numexpr\eq@tagpos@row@+\@ne@\relax
1141     \eq@tagpos@footroom@\eq@line@interline@
1142   \fi
1143 }%
1144 }
```

TODO: describe

```
1145 \def\eq@tagpos@print@line@eval{%
1146   \ifdefined\eq@tagpos@continuous
1147     \eq@tagpos@print@line@eval@continuous
1148   \else
1149     \eq@tagpos@print@line@eval@discrete
1150   \fi
1151 }
```

TODO: describe

```
1152 \def\eq@tagpos@print@line@eval@continuous{%
1153   \if@eqnsw
1154     \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@smashup@\relax
1155     \dp\eq@tagbox@\dimexpr\dp\eq@tagbox@-\eq@tagpos@smashdown@\relax
1156     \eq@tagpos@plain@\eq@tagpos@target@
1157     \@tempdima\dimexpr\eq@line@height@+\eq@tagpos@headroom@
1158       -\ht\eq@tagbox@\relax
1159     \@tempdimb\dimexpr-\eq@line@depth@-\eq@tagpos@footroom@
1160       +\dp\eq@tagbox@\relax
1161     \ifnum\eq@row@=\@ne
1162       \@tempdima.5\maxdimen
1163     \fi
1164     \ifnum\eq@row@=\eq@totalrows@
1165       \@tempdimb-.5\maxdimen
1166     \fi
1167     \ifdim\eq@tagpos@plain@>\@tempdima
```

```

1168     \ifdim\eql@tagpos@plain@>\@tempdimb
1169     \ifdim\@tempdima>\@tempdimb
1170         \eql@tagpos@plain@\@tempdima
1171     \else
1172         \eql@tagpos@plain@\@tempdimb
1173     \fi
1174 \fi
1175 \else
1176     \ifdim\eql@tagpos@plain@<\@tempdimb
1177     \ifdim\@tempdima>\@tempdimb
1178         \eql@tagpos@plain@\@tempdimb
1179     \else
1180         \eql@tagpos@plain@\@tempdima
1181     \fi
1182 \fi
1183 \fi
1184 \ifnum\eql@tagpos@prevrow@>\z@
1185     \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1186     \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1187         \eql@tagpos@raised@\eql@tagpos@plain@
1188         \let\eql@tagpos@reserve\eql@false
1189     \fi
1190 \else
1191     \ifdim\eql@tagpos@target@>%
1192         \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1193         \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1194     \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1195         \eql@tagpos@raised@\eql@tagpos@plain@
1196         \let\eql@tagpos@reserve\eql@false
1197     \fi
1198 \else
1199     \eql@tagpos@raised@\dimexpr-\eql@line@depth@
1200         -\ht\eql@tagbox@\relax
1201     \ifdim\eql@tagpos@raised@<\eql@tagpos@plain@\else
1202         \eql@tagpos@raised@\eql@tagpos@plain@
1203         \let\eql@tagpos@reserve\eql@false
1204     \fi
1205 \fi
1206 \fi
1207 \else
1208     \ifnum\eql@tagpos@prevrow@=\eql@row@
1209         \eql@tagwidth@\eql@tagwidth@block@
1210     \else
1211         \let\eql@tagpos@reserve\eql@false
1212     \fi
1213 \fi
1214 }

```

TODO: describe

```

1215 \def\eql@tagpos@print@line@eval@discretef%
1216 \if@eqnsw
1217     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1218     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1219     \eql@tagpos@plain@\eql@tagpos@shift@
1220     \eql@tagpos@headroom@\z@
1221     \eql@tagpos@footroom@\z@
1222     \ifdim\eql@tagpos@shift@>%
1223         \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi

```

```

1224     \eql@tagpos@raised@dimexpr\eq@line@height@+\dp\eq@tagbox@relax
1225     \else
1226     \eql@tagpos@raised@dimexpr-\eq@line@depth@-\ht\eq@tagbox@relax
1227     \fi
1228     \else
1229     \let\eq@tagpos@reserve\eq@false
1230     \fi
1231 }

```

TODO: describe

```

1232 \def\eq@tagpos@print@line@end{%
1233   \ifdefined\eq@tagpos@continuous
1234     \ifnum\eq@tagpos@prevrow@=\eq@row@
1235       \ifdefined\eq@tagpos@reserve
1236         \global\eq@appendexpand\eq@tags@container@block{%
1237           \advance\eq@tagpos@headroom@the\dimexpr\eq@line@height@
1238             +\eq@line@depth@relaxrelax}%
1239         \eq@displaybreak@star\M
1240       \fi
1241     \fi
1242   \fi
1243 }

```

E.6 Component Display

Showkeys Integration. **TODO:** describe

```

1244 \let\eq@SK@loaded\eq@false
1245 \let\eq@SK@label\@gobble
1246 \let\eq@SK@clearlabel\@empty
1247 \let\eq@SK@setlabel\@gobble
1248 \let\eq@SK@printlabel@right\@empty
1249 \let\eq@SK@printlabel@left\@empty
1250 \let\eq@SK@printlabel@line\@empty
1251 \def\eq@label@clean{\eq@label@org}
1252 \AddToHook{package/showkeys/after}{
1253   \let\eq@SK@loaded\eq@true
1254   \def\eq@SK@label#1{\SK@\SK@@label#1}
1255   \def\eq@SK@clearlabel{\let\eq@SK@lab\relax}
1256   \eq@SK@clearlabel
1257   \def\eq@SK@@label#1>#2\SK@{%
1258     \def\eq@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1259   }
1260   \def\eq@SK@setlabel#1{\SK@\eq@SK@@label#1}
1261   \def\eq@SK@printlabel@right{%
1262     \ifx\eq@SK@lab\relax\else
1263       \rlap{\kern\marginparsep\eq@SK@lab}%
1264       \eq@SK@clearlabel
1265     \fi
1266   }
1267   \def\eq@SK@printlabel@left{%
1268     \ifx\eq@SK@lab\relax\else
1269       \llap{\eq@SK@lab\kern\marginparsep}%
1270       \eq@SK@clearlabel
1271     \fi
1272   }
1273   \def\eq@SK@printlabel@line{%

```

```

1274 \ifx\eqL@SK@lab\relax\else
1275 \dimen@\prevdepth
1276 \nointerlineskip
1277 \ifdefined\eqL@tagsleft
1278 \llap{%
1279 \eqL@SK@lab
1280 \kern\marginparsep
1281 }%
1282 \eqL@SK@clearlabel
1283 \else
1284 \rlap{%
1285 \dimen@\displaywidth
1286 \advance\dimen@\marginparsep
1287 \kern\dimen@
1288 \eqL@SK@lab
1289 }%
1290 \fi
1291 \eqL@SK@clearlabel
1292 \prevdepth\dimen@
1293 \fi
1294 }
1295 \let\eqL@label@org\label
1296 \def\eqL@label@clean{\let\SK@\gobbletwo\eqL@label@org}
1297 }

```

Labels.

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

```

1298 \def\eqL@composetag@label{%
1299 \eqL@SK@clearlabel
1300 \ifdefined\eqL@tags@label
1301 \eqL@SK@setlabel\eqL@tags@label
1302 \ifdefined\eqL@tags@name
1303 \let\@currentlabelname\eqL@tags@name
1304 \else
1305 \let\@currentlabelname\eqL@tags@name@generic
1306 \fi
1307 \expandafter\eqL@label@clean\expandafter{\eqL@tags@label}%
1308 \fi
1309 }

```

TODO: describe

```

1310 \def\eqL@tags@printsubeqlabel{%
1311 \eqL@tags@container@parent
1312 \ifdefined\eqL@tags@label
1313 \eqL@tags@makeblockanchor
1314 \eqL@SK@setlabel\eqL@tags@label
1315 \begingroup
1316 \def\@currentcounter{equation}%
1317 \eqL@tags@anchor
1318 \let\@currentlabelname\eqL@tags@name@generic
1319 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1320 \expandafter\eqL@label@clean\expandafter{\eqL@tags@label}%
1321 \endgroup
1322 \eqL@SK@printlabel@line
1323 \fi
1324 }

```

Hyperref Anchors. **TODO:** describe

```
1325 \let\eqL@Hy@anchor\@gobble
1326 \AddToHook{package/hyperref/after}{
1327   \def\eqL@Hy@anchor#1{%
1328     \Hy@raisedlink{\hyper@anchor{#1}}%
1329   }%
1330 }
```

TODO: describe

```
1331 \def\eqL@tags@makeblockanchor{%
1332   \eqL@tags@glabel@step
1333   \eqL@Hy@anchor\eqL@tags@glabel
1334   \edef\eqL@tags@anchor{%
1335     \def\noexpand\thepage{\thepage}%
1336     \def\noexpand\@currentHref{\eqL@tags@glabel}%
1337   }%
1338 }
```

TODO: describe

`eqL@composetag@anchor`

```
1339 \def\eqL@composetag@anchor{%
1340   \ifdefined\eqL@tags@tag
1341     \def\@currentcounter{equation}%
1342     \ifdefined\eqL@tags@ref
1343       \let\@currentlabel\eqL@tags@ref
1344     \else
1345       \protected@edef\@currentlabel{\p@equation\eqL@tags@tag}%
1346     \fi
1347     \eqL@tags@glabel@step
1348     \edef\@currentHref{\eqL@tags@glabel}%
1349     \eqL@Hy@anchor\@currentHref
1350   \else
1351     \refstepcounter{equation}%
1352     \protected@edef\eqL@tags@tag{\theequation}%
1353   \fi
1354   \eqL@tags@anchor
1355 }
```

Tag Layout. **TODO:** describe

```
1356 \def\eqL@tags@taglayout@set@direct#1{%
1357   \def\eqL@tags@taglayout##1{#1}%
1358 }
1359 \def\eqL@tags@taglayout@set#1{%
1360   \def\eqL@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1361 }
```

TODO: describe

```
1362 \def\eqL@tags@tagform@set@direct#1{%
1363   \def\eqL@tags@tagform##1{#1}%
1364 }
1365 \def\eqL@tags@tagform@set#1#2#3{%
1366   \def\eqL@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1367 }
```



```

1368 \eql@tags@taglayout@set{#1}
1369 \eql@tags@tagform@set({#1})
1370 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1371 \protected\def\tagform{\eql@tags@tagform}
1372 \protected\def\tagbox{\eql@tags@taglayout}
1373 \protected\def\tagboxed{\eql@tags@tagcompose}

```

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

```

1374 \protected\def\eqleqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1375 \def\eql@composetag@tag{%
1376   \eql@tagging@tagbegin
1377   \eql@tags@frame@cmd{%
1378     \eql@tags@taglayout{%
1379       \eql@tags@tagform\eql@tags@tag
1380       \eql@tagging@tagsave
1381     }%
1382   }%
1383   \eql@tagging@tagend
1384 }

```

E.7 Tag Composition

TODO: describe

```

1385 \def\eql@composetag@measure{%
1386   \ifdefined\eql@tags@tag\else
1387     \stepcounter{equation}%
1388     \let\eql@tags@tag\theequation
1389   \fi
1390   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1391   \ifdefined\eql@numbering@multi
1392     \global\let\eql@tags@container\eql@tags@container@clear
1393   \fi
1394 }

```

TODO: describe

```

1395 \def\eql@composetag@print{%
1396   \eql@composetag@anchor
1397   \eql@composetag@label
1398   \ifdefined\eql@tags@left
1399     \eql@SK@printlabel@left
1400   \eql@composetag@tag
1401   \else
1402     \eql@composetag@tag
1403     \eql@SK@printlabel@right
1404   \fi
1405   \global\let\eql@tags@container\eql@tags@container@clear
1406 }

```

TODO: describe

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

```

1407 \def\eq@tagbox@make#1{%
1408   \setbox\eq@tagbox@\hbox{\eq@strut@tag@\@lign#1}%
1409   \eq@tagwidth@\wd\eq@tagbox@
1410   \ifdim\eq@tagwidth@<\eq@tagwidthmin@
1411     \eq@tagwidth@\eq@tagwidthmin@
1412   \fi
1413   \advance\eq@tagwidth@\eq@tagsepmin@
1414 }

```

TODO: describe

```

1415 \def\eq@tagbox@print@adjustheadroom{%
1416   \dimen@\dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1417   \ifdim\dimen@>\z@
1418     \ifdim\dimen@>\eq@tagpos@headroom@
1419       \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@headroom@\relax
1420     \else
1421       \ht\eq@tagbox@\dimexpr\eq@line@height@-\eq@tagpos@current@\relax
1422     \fi
1423   \fi
1424 }

```

TODO: describe

```

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

```

TODO: describe

```

1435 \def\eq@tagbox@print@extendabove{%
1436   \dimen@\dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1437   \ifdim\dimen@>\z@
1438     \global\eq@appendexpand\eq@display@container{%
1439       \eq@display@aboveextend@the\dimen@\relax}%
1440   \fi
1441 }

```

TODO: describe

```

1442 \def\eq@tagbox@print@extendbelow{%
1443   \dimen@\dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax
1444   \ifdim\dimen@>\z@
1445     \global\eq@appendexpand\eq@display@container{%
1446       \eq@display@belowextend@the\dimexpr\dimen@\relax}%
1447   \fi
1448 }

```

TODO: describe

```

1449 \def\eq@tagbox@print@prepare{%
1450   \ifdefined\eq@tagpos@reserve
1451     \eq@tagpos@current@\eq@tagpos@plain@
1452   \else
1453     \eq@tagpos@current@\eq@tagpos@raised@

```

```

1454 \fi
1455 \ifdim\eql@tagpos@headroom@>\z@
1456 \eql@tagbox@print@adjustheadroom
1457 \fi
1458 \ifdim\eql@tagpos@footroom@>\z@
1459 \eql@tagbox@print@adjustfootroom
1460 \fi
1461 \ifnum\eql@row@=\@ne
1462 \eql@tagbox@print@extendabove
1463 \fi
1464 \ifnum\eql@row@=\eql@totalrows@
1465 \eql@tagbox@print@extendbelow
1466 \fi
1467 }

```

TODO: describe

```

1468 \def\eql@tagbox@print@tagsright{%
1469 \eql@tagbox@print@prepare
1470 \kern-\wd\eql@tagbox@
1471 \raise\eql@tagpos@current@\box\eql@tagbox@
1472 }

```

TODO: describe

```

1473 \def\eql@tagbox@print@tagsleft{%
1474 \eql@display@firstavail@set\z@
1475 \eql@tagbox@print@prepare
1476 \wd\eql@tagbox@\z@
1477 \raise\eql@tagpos@current@\box\eql@tagbox@
1478 }

```

ql@tagbox@print@cell

```

1479 \def\eql@tagbox@print@cell{%
1480 \eql@tagging@tagaddbox
1481 \ifdefined\eql@tagsleft
1482 \ifdefined\eql@tagpos@reserve
1483 \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1484 \let\eql@tagpos@reserve\eql@false
1485 \fi
1486 \fi
1487 \if@eqnsw
1488 \eql@tagbox@print@tagsleft
1489 \fi
1490 \kern\displaywidth
1491 \else
1492 \kern\displaywidth
1493 \ifdefined\eql@tagpos@reserve
1494 \ifdim\eql@tagwidth@>%
1495 \dimexpr\displaywidth-\eql@line@width@+\eql@tagfuzz@\relax
1496 \let\eql@tagpos@reserve\eql@false
1497 \fi
1498 \fi
1499 \if@eqnsw
1500 \eql@tagbox@print@tagsright
1501 \fi
1502 \fi
1503 }

```

F Subequation Numbering

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

F.1 Definitions

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

```
1504 \eql@amsmath@undefine\c@parentequation
1505 \eql@amsmath@undefine\theparentequation
1506 \ifdefined\c@parentequation\else
1507 \newcounter{parentequation}
1508 \fi
```

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

```
1509 \def\eql@subequations@template{\theparentequation\alph{equation}}
```

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

```
1510 \let\eql@subequations@active\eql@false
```

`\eql@subequations@init` Low-level initialise the subequations mode. Store the equation counter in `\eql@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`:

```
1511 \def\eql@subequations@init{%
1512   \edef\eql@subequations@restorecounter{%
1513     \global\c@equation\the\c@equation\relax}%
1514   \eql@tags@container@block
1515   \ifdefined\eql@tags@tag
1516     \eql@tags@glabel@step
1517     \protected@edef\theHparentequation{\eql@tags@glabel}%
1518     \protected@edef\theparentequation{\eql@tags@tag}%
1519   \else
1520     \advance\c@equation\@ne
1521     \protected@edef\theparentequation{\theequation}%
1522     \ifdefined\theHequation
1523       \protected@edef\theHparentequation{\theHequation}%
1524     \fi
1525   \fi
1526   \global\c@parentequation\c@equation
1527   \global\c@equation\z@
1528   \let\theequation\eql@subequations@template
1529   \def\theHequation{\theHparentequation.\arabic{equation}}%
1530 }
```

`@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:

```

1531 \def\eq@subequations@close{%
1532   \ifnum\c@equation=\z@
1533     \eq@subequations@restorecounter
1534   \else
1535     \global\c@equation\c@parentequation
1536   \fi
1537 }
```

F.2 Environment

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

TODO: join with other similar anchor routines `\eq@tags@prints@sublabel`

```

1538 \def\eq@subequations@start{%
1539   \let\eq@tags@container@block\eq@tags@container@clear
1540   \eq@nextopt@process{subequations}%
1541   \eq@subequations@init
1542   \eq@tags@glabel@step
1543   \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1544   \eq@Hy@anchor\eq@subequations@currentHref
1545   \edef\eq@subequations@thepage{\thepage}%
1546   \def\@currentcounter{equation}%
1547   \let\@currentHref\eq@subequations@currentHref
1548   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1549   \eq@tags@container@block
1550   \ifdefined\eq@tags@name
1551     \let\@currentlabelname\eq@tags@name
1552   \else
1553     \let\@currentlabelname\eq@tags@name@generic
1554   \fi
1555   \let\eq@subequations@active\eq@true
1556   \ifdefined\eq@tags@label
1557     \eq@SK@label\eq@tags@label
1558   \fi
1559   \ignorespaces
1560 }
```

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

```

1561 \def\eq@subequations@end{%
1562   \ifnum\c@equation>\z@
1563     \eq@tags@container@block
1564     \ifdefined\eq@tags@label
1565       \begingroup
1566         \def\@currentcounter{equation}%
1567         \let\thepage\eq@subequations@thepage
1568         \let\@currentHref\eq@subequations@currentHref
1569 % \TODO how about tag* ?! also within equations!
1570         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1571         \ifdefined\eq@tags@name
1572           \let\@currentlabelname\eq@tags@name
1573         \else
1574           \let\@currentlabelname\eq@tags@name@generic

```

```

1575         \fi
1576         \expandafter\eqL@label@clean\expandafter{\eqL@tags@label}%
1577     \endgroup
1578     \fi
1579 \fi
1580 \eqL@subequations@close
1581 \ignorespacesafterend
1582 }

```

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

```

1583 \newenvironment{eqL@subequations}{%
1584 <dev>\eqL@dev@enterenv
1585   \eqL@subequations@testall\eqL@subequations@start%
1586 }{%
1587   \eqL@subequations@end
1588 <dev>\eqL@dev@leaveenv
1589 }

```

TODO: describe

```

1590 \def\eqL@subequations@testall{\eqL@parseopt\eqL@subequations@parseopt}
1591 \def\eqL@subequations@parseopt{%
1592   \ifx\eqL@parseopt@token[%]
1593     \let\eqL@parseopt@next\eqL@parseopt@opt
1594   \fi
1595   \ifx\eqL@parseopt@token\eqL@atxi
1596     \let\eqL@parseopt@next\eqL@parseopt@label
1597   \fi
1598   \ifx\eqL@parseopt@token\eqL@atxii
1599     \let\eqL@parseopt@next\eqL@parseopt@label
1600   \fi
1601   \ifx\eqL@parseopt@token\label
1602     \let\eqL@parseopt@next\eqL@parseopt@end
1603   \fi
1604 }

```

F.3 Subequation Scheme

TODO: describe

```

1605 \def\eqL@numbering@subeq@init{%
1606   \let\eqL@save@theequation\theequation
1607   \let\eqL@save@theHequation\theHequation
1608   \eqL@subequations@init
1609   \let\eqL@tags@container@parent\eqL@tags@container@block
1610   \let\eqL@tags@container@block\eqL@tags@container@clear
1611 }

```

TODO: describe

```

1612 \def\eqL@numbering@subeq@test{%
1613   \ifnum\eqL@tagrows@<\tw@
1614     \let\eqL@tags@container@block\eqL@tags@container@parent
1615     \let\eqL@numbering@subeq@use\eqL@false
1616     \let\theequation\eqL@save@theequation
1617     \let\theHequation\eqL@save@theHequation
1618     \eqL@subequations@restorecounter

```

```
1619 \fi
1620 }
```

TODO: describe

```
1621 % \TODO note must not use setcounter here (when calc is loaded)
1622 \def\eq@numbering@subeq@close{%
1623 \eq@subequations@close
1624 }
```

G Display Equations Support

TODO: describe

```
1625 \let\eq@display@injectbefore\@undefined
1626 \let\eq@display@injectafter\@undefined
1627 \let\eq@interline@container\@undefined
1628 \def\eq@interline@container@clear{%
1629 \eq@displaybreak@pen@\@MM
1630 \eq@vspaceskip@\z@skip
1631 }
```

G.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1632 \interdisplaylinepenalty\@M
```

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

```
1633 \def\eq@getdsp@pen#1{%
1634 \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1635 }
```

TODO: allow a displaybreak before equations

```
1636 \protected\def\eq@displaybreak@default{%
1637 \eq@warning{Invalid use of \string\displaybreak}{}}%
1638 \eq@teststaroropt@loose\@gobble\eq@gobbleopt{}}
1639 \eq@amsmath@after{\let\eq@displaybreak@default\displaybreak}
1640 \eq@amsmath@let\displaybreak\eq@displaybreak@default
```

```
1641 \newcount\eq@displaybreak@pen@
1642 \newcount\eq@displaybreak@prepen@
1643 \newcount\eq@displaybreak@postpen@
```

TODO: describe

```
1644 \protected\def\eq@displaybreak{%
1645 \relax
1646 \eq@ampprotecttwo\eq@teststaroropt@tight
1647 \eq@displaybreak@star\eq@displaybreak@level{4}}%
1648 }
```

```
1649 \def\eq@displaybreak@star#1{%
1650 \global\eq@appendexpand\eq@interline@container{%
```

```

1651 \eql@displaybreak@open@the\numexpr#1\relax\relax}%
1652 }

```

```

1653 \def\eql@displaybreak@level[#1]{%
1654 \ifnum#1<\z@
1655 \global\eql@append\eql@interline@container{\eql@displaybreak@open@\@MM}%
1656 \else
1657 \global\eql@appendexpand\eql@interline@container{%
1658 \eql@displaybreak@open@-\@getpen{#1}\relax}%
1659 \fi
1660 }

```

TODO: describe

```

1661 \def\eql@displaybreak@pre#1{%
1662 \ifnum#1<\z@
1663 \eql@displaybreak@prepen@\@MM
1664 \else
1665 \eql@displaybreak@prepen@-\@getpen{#1}\relax
1666 \fi
1667 }

```

TODO: describe

```

1668 \def\eql@displaybreak@post#1{%
1669 \ifnum#1<\z@
1670 \eql@displaybreak@postpen@\@MM
1671 \else
1672 \eql@displaybreak@postpen@-\@getpen{#1}\relax
1673 \fi
1674 }

```

TODO: describe

```

1675 \def\eql@displaybreak@inter#1{%
1676 \ifnum#1<\z@
1677 \interdisplaylinepenalty\@M
1678 \else
1679 \interdisplaylinepenalty\eql@getdsp@open{#1}\relax
1680 \fi
1681 }

```

G.2 Explicit Vertical Space

TODO: describe

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

```

1682 \newskip\eql@vspaceskip@

1683 \let\eql@vspace@org\vspace
1684 \def\eql@vspace{%
1685 \ifvmode
1686 \expandafter\eql@vspace@immediate
1687 \else
1688 \expandafter\eql@vspace@line
1689 \fi
1690 }

```


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

```

1691 \def\eql@vspace@line{%
1692   \eql@ifstar@loose\eql@vspace@addfixedbefore\eql@vspace@add
1693 }
1694 \def\eql@vspace@add#1{%
1695   \global\eql@appendexpand\eql@interline@container{%
1696     \advance\eql@vspaceskip@the\glueexpr#1\relax\relax}}
1697 \def\eql@vspace@addfixedbefore#1{%
1698   \global\eql@appendexpand\eql@interline@container{%
1699     \noexpand\eql@append\noexpand\eql@display@injectbefore{%
1700       \skip@the\glueexpr#1\relax\relax
1701       \penalty\@M
1702       \vskip\skip@
1703       \global\advance\eql@line@interline@\skip@
1704     }%
1705   }%
1706 }
1707 \def\eql@vspace@addfixedafter#1{%
1708   \global\eql@appendexpand\eql@interline@container{%
1709     \noexpand\eql@append\noexpand\eql@display@injectafter{%
1710       \dimen@\prevdepth
1711       \hrule\@height\z@
1712       \skip@the\glueexpr#1\relax\relax
1713       \penalty\@M
1714       \vskip\skip@
1715       \global\advance\eql@line@interline@\skip@
1716       \prevdepth\dimen@
1717     }%
1718   }%
1719 }

```

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

```

1720 \def\eql@vspace@immediate{%
1721   \noalign\bgroup
1722   \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
1723 }
1724 \def\eql@vspace@fixed#1{%
1725   \skip@\glueexpr#1\relax
1726   \ifnum\eql@row@=\@ne
1727     \global\eql@appendexpand\eql@display@container{%
1728       \advance\eql@abovespace@the\skip@\relax}%
1729   \else\ifnum\eql@row@>\eql@totalrows@
1730     \global\eql@appendexpand\eql@display@container{%
1731       \advance\eql@belowspace@the\skip@\relax}%
1732   \else
1733     \dimen@\prevdepth
1734     \hrule\@height\z@
1735     \penalty\@M
1736     \vskip\skip@
1737     \global\advance\eql@line@interline@\skip@
1738     \prevdepth\dimen@
1739   \fi\fi
1740   \egroup
1741 }
1742 \def\eql@vspace@discardable#1{%
1743   \skip@\glueexpr#1\relax

```

```

1744 \ifnum\eql@row@=\@ne
1745 \global\eql@appendexpand\eql@display@container{%
1746 \advance\eql@abovespace@the\skip@relax}%
1747 \else\ifnum\eql@row@>\eql@totalrows@
1748 \global\eql@appendexpand\eql@display@container{%
1749 \advance\eql@belowspace@the\skip@relax}%
1750 \else
1751 \vskip\skip@
1752 \global\advance\eql@line@interline@\skip@
1753 \fi\fi
1754 \egroup
1755 }

```

G.3 Default Vertical Spacing

TODO: describe

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

```

1756 \newbox\eql@strutbox@
1757 \def\eql@strut@depth{.3}
1758 \def\eql@strut{\copy\eql@strutbox@}
1759 \let\eql@strut@cell\eql@strut
1760 \let\eql@strut@tag\eql@strut
1761 \def\eql@strut@make{%
1762 \setbox\eql@strutbox@\hbox{%
1763 \tempdimb\dimexpr
1764 \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1765 \tempdima\dimexpr
1766 \normalbaselineskip-\normallineskiplimit-\tempdimb\relax
1767 \vrule\@height\tempdima\@depth\tempdimb\@width\z@
1768 }
1769 }
1770 \AtBeginDocument{\eql@strut@make}

```

TODO: describe

```

1771 \def\eql@spread@set{%
1772 \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
1773 +\normalbaselineskip-\baselineskip\relax
1774 \ifdim\eql@spread@>\z@
1775 \openup\eql@spread@
1776 \ifdefined\spread@equation
1777 \let\spread@equation\empty
1778 \fi
1779 \fi
1780 }

```

G.4 Entry and Exit

TODO: describe

```

1781 \let\eql@beamerbasecolor@fix\empty
1782 \AddToHook{package/beamerbasecolor/after}{%
1783 \def\eql@beamerbasecolor@fix{%
1784 \donotcolorouterdisplaymaths

```

```

1785 \donotcoloroutermaths
1786 \beamer@setdisplaymathcolor
1787 }%
1788 }

```

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

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

```

1789 \newskip\eql@abovespace@
1790 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

1791 \def\eql@display@enter{%
1792 \ifnoskipsec\leavevmode\par\fi
1793 \ifvmode
1794 \eql@prevdepth@\prevdepth
1795 \nointerlineskip
1796 \noindent
1797 \else
1798 \eql@prevdepth@\maxdimen
1799 \fi
1800 \eql@beamerbasecolor@fix
1801 }

```

`\eql@display@adjust`

```

1802 \def\eql@display@adjust{%
1803 \ifdefined\eql@display@linewidth
1804 \displaywidth\glueexpr\eql@display@linewidth\relax
1805 \advance\displaywidth-\displayindent
1806 \fi
1807 \ifdefined\eql@display@marginleft
1808 \advance\displaywidth\displayindent
1809 \displayindent\glueexpr\eql@display@marginleft\relax
1810 \advance\displaywidth-\displayindent
1811 \fi
1812 \ifdefined\eql@display@marginright
1813 \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1814 \fi
1815 \ifdim\displaywidth<\z@
1816 \displaywidth\z@
1817 \fi
1818 }

```

`\eql@display@init`

```

1819 \def\eql@display@init{%
1820 \let\displaybreak\eql@displaybreak
1821 \let\eql@vspace@org\vspace
1822 \let\vspace\eql@vspace
1823 \let\eqncontrol\eql@control
1824 \let\eql@display@injectbefore\@empty
1825 \let\eql@display@injectafter\@empty
1826 \eql@spread@set
1827 \eql@strut@make
1828 \let\eql@frame@cmd\@undefined
1829 }

```

`\eql@display@print`

```
1830 \def\eql@display@print{%
1831   \let\eql@display@container\@empty
1832   \eql@display@firstavail@z@
1833   \eql@display@aboveextend@z@
1834   \eql@display@belowextend@z@
1835   \global\let\eql@interline@container\eql@interline@container@clear
1836 }
```

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

```
1837 \def\eql@display@halign@init#1{%
1838   \eql@row@z@
1839   \eql@prevgraf@\prevgraf
1840   \everycr{\noalign{%
1841     \global\advance\eql@row@\@ne
1842     \prevgraf\numexpr\prevgraf+\@ne\relax
1843     #1%
1844   }}%
1845 }
```

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

```
1846 \def\eql@display@halign@start{%
1847   \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1848   \ifdim\eql@prevdepth@=\maxdimen\else
1849     \prevdepth\eql@prevdepth@
1850   \fi
1851   \ifdim\prevdepth=-\@m\p\else
1852     \ifdefined\eql@display@height
1853       \skip@\baselineskip
1854       \advance\skip@-\glueexpr\eql@display@height\relax
1855       \advance\skip@-\prevdepth\relax
1856       \ifdim\skip@<\lineskiplimit
1857         \skip@\lineskip
1858       \fi
1859       \advance\skip@-\eql@spread@\relax
1860       \vskip\skip@
1861       \nointerlineskip
1862     \else
1863       \vskip-\eql@spread@\relax
1864     \fi
1865   \fi
1866 }
```

TODO: describe

```
1867 \def\eql@display@vspace{%
1868   \advance\abovedisplayskip\eql@abovespace@
1869   \advance\belowdisplayskip\eql@belowspace@
1870 }
```

TODO: describe

```
1871 \def\eql@display@vspace@native{%
1872   \advance\abovedisplayskip\eql@abovespace@
1873   \advance\belowdisplayskip\eql@belowspace@
1874   \advance\abovedisplayshortskip\eql@abovespace@
1875   \advance\belowdisplayshortskip\eql@belowspace@
1876 }
```

TODO: describe

```
1877 \def\eqldisplay@penalty{%
1878   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1879     \postdisplaypenalty\eqldisplaybreak@postpen@
1880   \fi
1881   \ifnum\eqldisplaybreak@open@=\@MM\else
1882     \postdisplaypenalty\eqldisplaybreak@open@
1883   \fi
1884   \ifnum\eqldisplaybreak@prepen@=\@MM\else
1885     \predisplaypenalty\eqldisplaybreak@prepen@
1886   \fi
1887 }
```

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

```
1888 \def\eqldisplay@halign@end{%
1889   \eql@interline@container
1890   \eqldisplay@injectbefore
1891   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
1892   \ifdefined\eqldisplay@depth
1893     \prevdepth\glueexpr\eqldisplay@depth\relax
1894   \fi
1895 }
```

`\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?

```
1896 \def\eqldisplay@close{%
1897   \eqldisplay@container
1898   \ifdim\eqldisplay@firstavail@<\z@
1899     \eqldisplay@firstavail@\z@
1900   \fi
1901   \eql@skip@mode@leave@\z@
1902   \ifdim\eql@prevdepth@=\maxdimen
1903     \ifdim\predisplaysize=-\maxdimen
1904       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1905       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1906     \else
1907       \eql@skip@mode@above@\z@
1908       \eql@skip@mode@below@\z@
1909       \advance\eqldisplay@firstavail@\displayindent
1910       \ifdim\eqldisplay@firstavail@>\predisplaysize
1911         \ifcase\eql@skip@mode@short\relax
1912           \or
1913             \eql@skip@mode@above@\@ne
1914           \or
1915             \eql@skip@mode@above@\@ne
1916             \ifnum\eql@totalrows@=\@ne
1917               \eql@skip@mode@below@\@ne
1918             \fi
1919           \or
1920             \eql@skip@mode@above@\@ne
1921             \eql@skip@mode@below@\@ne
1922           \fi
1923         \fi
1924       \fi
1925     \else
1926       \ifdim\eql@prevdepth@=-\@m\p@
```

```

1927     \eql@skip@mode@above@\eql@skip@mode@top@above\relax
1928     \eql@skip@mode@below@\eql@skip@mode@top@below\relax
1929     \else
1930     \eql@skip@mode@above@\eql@skip@mode@par@above\relax
1931     \eql@skip@mode@below@\eql@skip@mode@par@below\relax
1932     \fi
1933 \fi
1934 \ifcase\eql@skip@mode@above@
1935 \or\or\or
1936     \predisplaypenalty\z@
1937 \or
1938     \predisplaypenalty\z@
1939 \fi
1940 \ifcase\eql@skip@mode@below@
1941 \or\or\or
1942     \eql@skip@mode@leave@\@ne
1943 \or
1944     \eql@skip@mode@leave@\tw@
1945 \fi
1946 \ifdefined\eql@skip@force@above
1947     \eql@skip@mode@above@\eql@skip@force@above\relax
1948 \fi
1949 \ifdefined\eql@skip@force@below
1950     \eql@skip@mode@below@\eql@skip@force@below\relax
1951 \fi
1952 \ifdefined\eql@skip@force@leave
1953     \eql@skip@mode@leave@\eql@skip@force@leave\relax
1954 \fi
1955 \ifnum\eql@skip@mode@leave@>\z@
1956     \postdisplaypenalty\z@
1957 \fi
1958 \ifcase\eql@skip@mode@above@
1959     \abovedisplayskip\glueexpr\eql@skip@long@above\relax
1960 \or
1961     \abovedisplayskip\glueexpr\eql@skip@short@above\relax
1962 \or
1963     \abovedisplayskip\glueexpr\eql@skip@cont@above\relax
1964 \or
1965     \abovedisplayskip\glueexpr\eql@skip@par@above\relax
1966 \or
1967     \abovedisplayskip\glueexpr\eql@skip@top@above\relax
1968 \or
1969     \abovedisplayskip\z@skip
1970 \or
1971     \abovedisplayskip\glueexpr\eql@skip@med@above\relax
1972 \or
1973     \abovedisplayskip\glueexpr\eql@skip@custom@above\relax
1974 \fi
1975 \ifcase\eql@skip@mode@below@
1976     \belowdisplayskip\glueexpr\eql@skip@long@below\relax
1977 \or
1978     \belowdisplayskip\glueexpr\eql@skip@short@below\relax
1979 \or
1980     \belowdisplayskip\glueexpr\eql@skip@cont@below\relax
1981 \or
1982     \belowdisplayskip\glueexpr\eql@skip@par@below\relax
1983 \or
1984     \belowdisplayskip\glueexpr\eql@skip@top@below\relax

```

```

1985 \or
1986   \belowdisplayskip\z@skip
1987 \or
1988   \belowdisplayskip\glueexpr\eq@skip@med@below\relax
1989 \or
1990   \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
1991 \fi
1992 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
1993 \eq@interline@container
1994 \advance\eq@belowspace@\eq@vspaceskip@
1995 \eq@display@penalty
1996 \eq@display@vspace
1997 \skip@\glueexpr\eq@skip@tag@above\relax
1998 \ifdim\skip@>\abovedisplayskip
1999   \skip@\abovedisplayskip
2000 \fi
2001 \advance\abovedisplayskip-\eq@display@aboveextend@\relax
2002 \ifdim\abovedisplayskip<\skip@
2003   \abovedisplayskip\skip@
2004 \fi
2005 \skip@\glueexpr\eq@skip@tag@below\relax
2006 \ifdim\skip@>\belowdisplayskip
2007   \skip@\belowdisplayskip
2008 \fi
2009 \ifdim\eq@display@belowextend@>\z@
2010   \advance\belowdisplayskip-\eq@display@belowextend@\relax
2011   \ifdim\belowdisplayskip<\skip@
2012     \belowdisplayskip\skip@
2013   \fi
2014 \fi
2015 }

```

TODO: describe

```

2016 \def\eq@display@leave{%
2017   \prevgraf\eq@prevgraf@
2018   \ifcase\eq@skip@mode@leave@
2019     \or
2020     \endgraf
2021     \or
2022     \endgraf
2023     \prevdepth-\@m\p@
2024   \fi
2025 }

```

TODO: describe

```

2026 \def\eq@display@nest{%
2027   \let\displaybreak\eq@displaybreak@default
2028   \let\intertext\eq@intertext@default
2029   \let\vspace\eq@vspace@org
2030 }

```

TODO: describe

```

2031 \def\eq@display@restore{%
2032   \let\label\eq@label@org
2033   \let>tag\eq@tag@default
2034   \let\raisetag\eq@raisetag@default
2035   \let\displaybreak\eq@displaybreak@default
2036   \let\intertext\eq@intertext@default

```

```
2037 \let\vspace\eq@vspace@org
2038 }
```

TODO: describe

```
2039 \eq@append@arrayparboxrestore{%
2040 \eq@display@restore
2041 \ifdefined\eq@ampproof@active
2042 \eq@amprevert
2043 \fi
2044 \@displayfalse
2045 }
```

G.5 Stack

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

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

```
2046 \def\eq@stack@enable{%
2047 \let\eq@stack@save@equations\eq@stack@save@equations@
2048 \let\eq@stack@save@box\eq@stack@save@box@
2049 }
```

TODO: describe

```
2050 \let\eq@stack@save@equations\eq@stack@enable
2051 \let\eq@stack@save@box\eq@stack@enable
2052 \let\eq@stack@restore\@empty
```

TODO: describe

```
2053 \def\eq@stack@save@reg#1{\global#1\the#1\relax}
2054 \def\eq@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

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

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

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

```
2055 \def\eq@stack@save@equations@{%
2056 \let\eq@stack@numbering@eqnswinit\eq@numbering@eqnswinit
2057 \let\eq@stack@cell@container\eq@cell@container
2058 \let\eq@stack@tags@container\eq@tags@container
2059 \let\eq@stack@interline@container\eq@interline@container
2060 \let\eq@stack@block@container\eq@display@container
2061 \let\eq@stack@dimensions@tab\eq@dimensions@tab
2062 \edef\eq@stack@restore{%
2063 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2064 \eq@stack@save@let\eq@stack@numbering@eqnswinit\eq@numbering@eqnswinit
2065 \eq@stack@save@let\eq@stack@cell@container\eq@cell@container
2066 \eq@stack@save@let\eq@stack@tags@container\eq@tags@container
2067 \eq@stack@save@let\eq@stack@interline@container\eq@interline@container
2068 \eq@stack@save@let\eq@stack@dimensions@tab\eq@dimensions@tab
2069 \eq@stack@save@let\eq@stack@block@container\eq@display@container
2070 \eq@stack@save@reg\eq@column@
2071 \eq@stack@save@reg\eq@totalcolumns@
2072 \eq@stack@save@reg\eq@line@avail@
2073 \eq@stack@save@reg\eq@line@pos@
2074 \eq@stack@save@reg\eq@line@width@
```



```

2075 \eql@stack@save@reg\eql@line@depth@
2076 \eql@stack@save@reg\eql@line@height@
2077 \eql@stack@save@reg\eql@line@prevdepth@
2078 \eql@stack@save@reg\eql@line@interline@
2079 \eql@stack@save@reg\eql@totalheight@
2080 \eql@stack@save@reg\eql@tagwidth@max@
2081 \eql@stack@save@reg\eql@tagpos@row@
2082 \eql@stack@save@reg\eql@row@
2083 \eql@stack@save@reg\eql@tagrows@
2084 }%
2085 }

```

TODO: describe

```

2086 \def\eql@stack@save@box@{%
2087 \let\eql@stack@cell@container\eql@cell@container
2088 \edef\eql@stack@restore@{%
2089 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2090 \eql@stack@save@reg\eql@row@
2091 }%
2092 }

```

H Multi-Line Support

TODO: describe

H.1 Measure Support

TODO: describe

```

2093 \def\eql@measure@init#1#2{%
2094 \eql@dimensions@reset
2095 \let\eql@display@container\@empty
2096 \eql@numbering@measure@init
2097 \eql@row@z@
2098 \eql@totalheight@z@
2099 \eql@totalrows@M
2100 \eql@line@prevdepth@-@m@p@
2101 \eql@line@interline@z@
2102 \tabskipz@skip
2103 \everycrf\@noalign{%
2104 \global\advance\eql@row@\@ne
2105 #1%
2106 }%
2107 \global\let\eql@interline@container\eql@interline@container@clear
2108 \eql@measure@savestate
2109 \eql@display@halign@letcr{#2}%
2110 }

```

TODO: describe

```

2111 \def\eql@measure@tag@{%
2112 \eql@tagwidth@z@
2113 \ifdefined\eql@numbering@multi
2114 \if@eqnsw
2115 \eql@tags@container
2116 \eql@tagbox@make\eql@composetag@measure

```

```

2117     \ifdefined\eq\tagpos@reserve\else
2118     \eq\tagwidth@z@
2119     \fi
2120 \fi
2121 \fi
2122 }

```

TODO: describe

```

2123 \def\eq\measure@endrow{%
2124 \ifdim\eq\line@prevdepth@=-\m\p@ \else
2125 \dimen@\dimexpr\baselineskip-\eq\line@height@-\eq\line@prevdepth@\relax
2126 \ifdim\dimen@<\lineskiplimit
2127 \dimen@\lineskip
2128 \fi
2129 \advance\eq\line@interline@\dimen@
2130 \fi
2131 \eq\dimensions@endrow
2132 \ifdim\eq\tagwidth@>\eq\tagwidth@max@
2133 \global\eq\tagwidth@max@\eq\tagwidth@
2134 \fi
2135 \ifdim\eq\tagwidth@>z@
2136 \global\advance\eq\tagrows@\@ne
2137 \fi
2138 \global\advance\eq\totalheight@\dimexpr
2139 \eq\line@interline@+\eq\line@height@+\eq\line@depth@
2140 \global\eq\line@interline@z@
2141 \global\eq\line@prevdepth@\eq\line@depth@
2142 }

```

TODO: describe

```

2143 \def\eq\measure@close{%
2144 \advance\eq\row@-\tw@
2145 \eq\totalrows@\eq\row@
2146 \ifnum\eq\totalrows@>z@
2147 \eq\dimensions@get@\@ne
2148 \eq\topheight@\dimexpr\eq\line@height@+\eq\line@interline@\relax
2149 \eq\dimensions@get\eq\totalrows@
2150 \eq\bottomdepth@\eq\line@depth@
2151 \fi
2152 \eq\numbering@measure@blocktag
2153 \begingroup
2154 \eq\tags@container
2155 \if@eqnsw
2156 \eq\tagbox@make\eq\composetag@measure
2157 \ifdefined\eq\tagpos@reserve\else
2158 \eq\tagwidth@z@
2159 \fi
2160 \eq\dimensions@saveblocktag
2161 \else
2162 \eq\dimensions@savenoblocktag
2163 \eq\numbering@warnunused
2164 \fi
2165 \endgroup
2166 \eq\dimensions@getz@
2167 \eq\measure@restorestate
2168 }

```

```

measure@restorestate
\let\@measure@savestate
2169 \let\@measure@restorestate\@empty
2170 \def\@measure@savestate{%
2171   \begingroup
2172     \def\@elt##1{%
2173       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2174     \global\edef\@gtempa{\cl@ckpt}%
2175   \endgroup
2176   \let\@measure@restorestate\@gtempa
2177 }

```

H.2 Line Breaks

TODO: describe

\@display@cr

```

2178 \protected\def\@display@cr{%
2179   \@ampprotecttwo\@teststaropt@tight{%
2180     \global\@append\@interline@container{\@displaybreak@pen@\MM}%
2181     \@display@cr@opt}
2182   \@display@cr@opt\z@skip
2183 }

```

\@display@cr@opt

```

2184 \def\@display@cr@opt[#1]{%
2185   \@display@endline
2186   \cr
2187   \noalign{%
2188     \@interline@container
2189     \@display@injectbefore
2190     \ifnum\@displaybreak@pen@=\MM
2191       \penalty\interdisplaylinepenalty
2192     \else
2193       \penalty\@displaybreak@pen@
2194     \fi
2195     \advance\@vspaceskip@\glueexpr#1\relax
2196     \vskip\@vspaceskip@
2197     \global\advance\@line@interline@\@vspaceskip@
2198     \@display@injectafter
2199     \global\let\@interline@container\@interline@container@clear
2200   }%
2201 }

```

\@display@halign@letcr

```

2202 \def\@display@halign@letcr#1{%
2203   \let\\\@display@cr
2204   \let\@display@endline#1%
2205 }

```

H.3 Intertext

TODO: describe

TODO: revert in everymath?

```
2206 \def\eq@intertext@default{\eq@error{Invalid use of \string\intertext}}
2207 \eq@amsmath@let\intertext\eq@intertext@default
```

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

```
2208 \def\eq@intertext@process{%
2209   \eq@display@endline
2210   \cr
2211   \ifmeasuring@
2212     \expandafter\@gobble
2213   \else
2214     \expandafter\eq@intertext@print
2215   \fi
2216 }
```

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

```
2217 \def\eq@intertext@print#1{%
2218   \noalign{%
2219     \eq@display@halign@end
2220     \let\eq@skip@force@below\z@
2221     \let\eq@skip@force@above\z@
2222     \eq@setkeys{intertext}\eq@intertext@opt
2223     \openup-\eq@spread@
2224     \penalty\postdisplaypenalty
2225     \ifcase\eq@skip@force@below\relax
2226       \advance\eq@vspaceskip@\glueexpr\eq@skip@long@below\relax
2227     \or
2228       \advance\eq@vspaceskip@\glueexpr\eq@skip@short@below\relax
2229     \or
2230       \advance\eq@vspaceskip@\glueexpr\eq@skip@cont@below\relax
2231     \or
2232       \advance\eq@vspaceskip@\glueexpr\eq@skip@par@below\relax
2233     \or
2234       \advance\eq@vspaceskip@\glueexpr\eq@skip@top@below\relax
2235     \or
2236       \advance\eq@vspaceskip@\z@skip
2237     \or
2238       \advance\eq@vspaceskip@\glueexpr\eq@skip@med@below\relax
2239     \or
2240       \advance\eq@vspaceskip@\glueexpr\eq@skip@custom@below\relax
2241     \fi
2242     \vskip\eq@vspaceskip@
2243     \global\let\eq@interline@container\eq@interline@container@clear
2244     \vbox{%
2245       \@parboxrestore
2246       \ifdim
2247         \ifdim@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2248       \else
2249         \parshape@ne
2250         \@totalleftmargin\linewidth
2251       \fi
2252       \noindent
2253       \prevgraf\eq@prevgraf@
2254       \ignorespaces
2255       #1%
2256     \par
```

```

2257     \global\eq@prevgraf@\prevgraf
2258   }%
2259   \penalty\predisplaypenalty
2260   \ifcase\eq@skip@force@above\relax
2261     \vskip\glueexpr\eq@skip@long@above\relax
2262   \or
2263     \vskip\glueexpr\eq@skip@short@above\relax
2264   \or
2265     \vskip\glueexpr\eq@skip@cont@above\relax
2266   \or
2267     \vskip\glueexpr\eq@skip@par@above\relax
2268   \or
2269     \vskip\glueexpr\eq@skip@top@above\relax
2270   \or
2271     \vskip\z@skip
2272   \or
2273     \vskip\glueexpr\eq@skip@med@above\relax
2274   \or
2275     \vskip\glueexpr\eq@skip@custom@above\relax
2276   \fi
2277 %   \eq@prevdepth@\maxdimen
2278   \eq@prevdepth@\z@
2279   \eq@display@halign@start
2280 }
2281 }

```

TODO: describe

```

2282 \newenvironment{eq@intertext}{%
2283   \eq@testopt@tight\eq@intertext@{}%
2284 }{%
2285   \aftergroup\eq@intertext@after
2286   \ignorespacesafterend
2287 }

```

TODO: describe

```

2288 \def\eq@intertext@env{intertext}
2289 \def\eq@intertext@[#1]{%
2290   \global\def\eq@intertext@opt{#1}%
2291   \ifx\@currenvir\eq@intertext@env
2292     \expandafter\eq@scan@env\expandafter\eq@intertext@inject
2293   \else
2294     \expandafter\eq@intertext@process
2295   \fi
2296 }

```

TODO: describe

```

2297 \def\eq@intertext@inject{%
2298   \global\edef\eq@intertext@after{%
2299     \noexpand\eq@intertext@process{%
2300       \ifx\eq@scan@body\eq@scan@body@dump
2301         \eq@scan@body@dump
2302       \else
2303         \noexpand\scantokens{\eq@scan@body@dump}%
2304       \fi
2305     }%
2306   }%
2307 }

```

H.4 Line Marks

TODO: describe

```
2308 \def\eq@markline@pos@below{below}
2309 \def\eq@markline@pos@bottom{bottom}
2310 \def\eq@markline@pos@baseline{baseline}
2311 \let\eq@markline@pos\eq@markline@pos@baseline
2312 \let\eq@markline@shift\z@
2313 \def\eq@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2314 \def\eq@markline@symbol{}
```

TODO: describe

```
2315 \def\eq@markline@select#1{%
2316   \let\eq@markline@shift\z@
2317   \eq@setkeys{markline}{#1}%
2318   \eq@markline@print
2319 }
```

TODO: describe

```
2320 \def\eq@markline@print{%
2321   \dimen@ \dimexpr\eq@markline@shift\relax
2322   \ifx\eq@markline@pos\eq@markline@pos@below
2323     \ifdim\dimen@=\z@\else
2324       \penalty\@M
2325       \vskip-\dimen@
2326       \fi
2327       \nointerlineskip
2328       \penalty\@M
2329       \vbox{\hfill\hbox{\eq@markline@symbol}}%
2330   \else
2331     \ifx\eq@markline@pos\eq@markline@pos@baseline
2332       \advance\dimen@\prevdepth
2333       \fi
2334       \setbox\z@\hbox{\raise\dimen@\hbox{\eq@markline@symbol}}%
2335       \dimen@\prevdepth
2336       \ht\z@\z@
2337       \dp\z@\z@
2338       \nointerlineskip
2339       \penalty\@M
2340       \vbox{\hfill\box\z@}%
2341       \prevdepth\dimen@
2342   \fi
2343 }
```

TODO: describe

```
2344 \def\eq@markline@inject#1{%
2345   \let\eq@markline@push\eq@false
2346   \ifx\eq@markline@pos\eq@markline@pos@below\else
2347     \ifdefined\eq@tagsleft\else
2348       \ifx\eq@equations@main\eq@multi@main
2349         \ifdefined\eq@numbering@multi
2350           \if@eqnsw
2351             \let\eq@markline@push\eq@true
2352           \fi
2353         \else
2354           \ifnum\eq@row@=\eq@tagpos@row@
```

```

2355         \let\eq\markline@push\eq@true
2356         \fi
2357     \fi
2358 \else
2359     \if@eqnsw
2360         \let\eq\markline@push\eq@true
2361     \fi
2362 \fi
2363 \fi
2364 \fi
2365 \ifdefined\eq\markline@push
2366     \global\eq\append\eq\interline@container{%
2367         \eq\append\eq\display@injectbefore{\eq\markline@select{push,#1}}}%
2368 \else
2369     \global\eq\append\eq\interline@container{%
2370         \eq\append\eq\display@injectbefore{\eq\markline@select{#1}}}%
2371 \fi
2372 }

```

TODO: describe

```

2373 \def\eq\markline@amsthm@opt[#1]{\eq\markline@inject{qed,#1}}
2374 \def\eq\markline@amsthm@staropt[#1]{\eq\markline@inject{qed,push,#1}}
2375 \def\eq\markline@amsthm@qed{\eq@teststaropt@tight
2376     \eq\markline@amsthm@staropt\eq\markline@amsthm@opt{}}
2377 \def\eq\markline@amsthm@register#1{\eq\letcs{#1@qed}\eq\markline@amsthm@qed}

```

I Column Placement

TODO: describe

I.1 Supporting Definitions

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

```

2378 \newcount\eq@shape@pos@
2379 \newdimen\eq@shape@amount@
2380 \let\eq@shape@lastrow\eq@false

```

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

$\eq@marginright@$ (*dimen*)
 $\eq@centeroffset@$ (*dimen*)

```

2381 \newdimen\eq@marginleft@
2382 \newdimen\eq@marginright@
2383 \newdimen\eq@marginleft@min@
2384 \newdimen\eq@centeroffset@

```

I.2 Shape Schemes

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

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

```

2385 \def\eq@shape@tab@default{default}
2386 \def\eq@shape@tab@left{left}
2387 \def\eq@shape@tab@center{center}
2388 \def\eq@shape@tab@right{right}
2389 \def\eq@shape@tab@first{first}
2390 \def\eq@shape@tab@hanging{hanging}
2391 \def\eq@shape@tab@steps{steps}

```

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

```

2392 \let\eq@shape@tab@def\eq@shape@tab@default
2393 \let\eq@shape@tab@\eq@shape@tab@default
2394 \let\eq@shape@tab@l\eq@shape@tab@left
2395 \let\eq@shape@tab@c\eq@shape@tab@center
2396 \let\eq@shape@tab@r\eq@shape@tab@right
2397 \let\eq@shape@tab@rc\eq@shape@tab@first
2398 \let\eq@shape@tab@indent\eq@shape@tab@first
2399 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2400 \let\eq@shape@tab@lc\eq@shape@tab@hanging
2401 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2402 \let\eq@shape@tab@lcr\eq@shape@tab@steps

```

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

```

2403 \let\eq@shape@mode\eq@shape@tab@default

```

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

```

2404 \def\eq@shape@set#1{%
2405   \ifcsname eq@shape@tab@#1\endcsname
2406     \expandafter\let\expandafter\eq@shape@mode
2407     \csname eq@shape@tab@#1\endcsname
2408   \else
2409     \eq@error{shape '#1' unknown: setting to default}%
2410     \let\eq@shape@mode\eq@shape@tab@default
2411   \fi
2412 }

```

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

```

2413 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@\eq@shape@amount@z@}
2414 \def\eq@shape@layoutcenter@center{\eq@shape@pos@\@ne\eq@shape@amount@z@}
2415 \def\eq@shape@layoutcenter@right{\eq@shape@pos@\tw@\eq@shape@amount@z@}
2416 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2417 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@\eq@shape@amount@z@}
2418 \def\eq@shape@layoutleft@center{\eq@shape@pos@\@ne\eq@shape@amount@z@}
2419 \def\eq@shape@layoutleft@right{\eq@shape@pos@\tw@\eq@shape@amount@z@}
2420 \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):

```

2421 \def\eq@shape@layoutcenter@first{%
2422   \eq@shape@pos@z@
2423   \eq@shape@amount@z@
2424   \ifnum\eq@totalrows@>\@ne
2425     \ifnum\eq@row@=\@ne
2426       \eq@shape@amount@\eq@indent@

```



```

2427   \fi
2428   \fi
2429 }
2430 \def\eq@shape@layout@left@first{%
2431   \eq@shape@pos@z@
2432   \eq@shape@amount@z@
2433   \ifnum\eq@totalrows@>\@ne
2434     \ifnum\eq@row@=\@ne
2435       \eq@shape@amount@\eq@indent@
2436     \fi
2437   \fi
2438 }

```

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:

```

2439 \def\eq@shape@layout@center@hanging{%
2440   \eq@shape@pos@z@
2441   \eq@shape@amount@\eq@indent@
2442   \ifnum\eq@totalrows@>\@ne
2443     \ifnum\eq@row@=\@ne
2444       \eq@shape@amount@z@
2445     \fi
2446   \fi
2447 }
2448 \def\eq@shape@layout@left@hanging{%
2449   \eq@shape@pos@z@
2450   \eq@shape@amount@z@
2451   \ifnum\eq@totalrows@>\@ne
2452     \ifnum\eq@row@=\@ne
2453       \eq@shape@amount@-\eq@indent@
2454     \fi
2455   \fi
2456 }

```

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:

```

2457 \def\eq@shape@layout@center@steps{%
2458   \eq@shape@amount@z@
2459   \eq@shape@pos@\@ne
2460   \ifnum\eq@totalrows@>\@ne
2461     \ifnum\eq@row@=\@ne
2462       \eq@shape@pos@z@
2463     \fi
2464     \ifnum\eq@row@=\eq@totalrows@
2465       \eq@shape@pos@tw@
2466     \fi
2467   \fi
2468 }
2469 \def\eq@shape@layout@left@steps{%
2470   \eq@shape@pos@z@
2471   \eq@shape@amount@z@
2472   \ifnum\eq@totalrows@>\@ne
2473     \ifnum\eq@row@=\@ne
2474       \eq@shape@amount@-\eq@indent@
2475     \fi

```

```

2476 \ifnum\eq@row@=\eq@totalrows@
2477 \eq@shape@amount@\eq@indent@
2478 \fi
2479 \fi
2480 }

```

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

```

2481 \let\eq@shape@eval@undefined
2482 \def\eq@shape@select{%
2483 \expandafter\let\expandafter\eq@shape@eval
2484 \csname eq@shape%
2485 @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2486 @\eq@shape@mode\endcsname
2487 }

```

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

`\eq@shape@alignright`

`\eq@shape@aligncenter`

```

2488 \protected\def\eq@shape@alignleft{%
2489 \global\eq@append\eq@cell@container{\eq@shape@pos@\z@}%
2490 \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2491 \protected\def\eq@shape@aligncenter{%
2492 \global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}%
2493 \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2494 \protected\def\eq@shape@alignright{%
2495 \global\eq@append\eq@cell@container{\eq@shape@pos@\tw@}%
2496 \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2497 \def\eq@shape@align@testpar#1{%
2498 \eq@ifstar@tight{#1[\eq@indent@]}%
2499 {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2500 {\eq@testopt@tight{#1}\z@}}
2501 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2502 \protected\def\eq@shape@alignamount{%
2503 \eq@ampprotecttwo\eq@ifstar@tight
2504 \eq@shape@alignamount@set\eq@shape@alignamount@add}
2505 \def\eq@shape@alignamount@add#1{%
2506 \global\eq@appendexpand\eq@cell@container{%
2507 \advance\eq@shape@amount@\the\glueexpr#1\relax\relax}}
2508 \def\eq@shape@alignamount@set#1{%
2509 \global\eq@appendexpand\eq@cell@container{%
2510 \eq@shape@amount@\the\glueexpr#1\relax\relax}}
2511 \def\eq@shape@align@enable{%
2512 \let\shoveleft\eq@shape@alignleft
2513 \let\shovecenter\eq@shape@aligncenter
2514 \let\shoveright\eq@shape@alignright
2515 \let\shoveby\eq@shape@alignamount
2516 }

```

TODO: describe

```

2517 \protected\def\eq@shape@align@default{%
2518 \eq@warn@here{\shove...}}%
2519 \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2520 \protected\def\eq@shape@alignamount@default{%

```

```

2521 \eql@warn@here{\shove...}%
2522 \eql@ampprotecttwo\eql@ifstar@tight@gobble@gobble}
2523 \def\eql@shape@align@disable{%
2524 \let\shoveleft\eql@shape@align@default
2525 \let\shovecenter\eql@shape@align@default
2526 \let\shoveright\eql@shape@align@default
2527 \let\shoveby\eql@shape@alignamount@default
2528 }

```

I.3 Width Data

`width@block@` (*dimen*)

```

2529 \newdimen\eql@tagwidth@block@
2530 \newdimen\eql@tagheight@block@
2531 \newdimen\eql@tagdepth@block@

```

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

```

2532 \let\eql@dimensions@tab@empty

```

`eql@dimensions@reset`

```

2533 \def\eql@dimensions@reset{%
2534 \let\eql@dimensions@tab@empty
2535 \eql@tagwidth@max@z@
2536 \eql@tagrows@z@
2537 }

```

`\eql@dimensions@add`

```

2538 \def\eql@dimensions@add#1{%
2539 \global\eql@appendexpand\eql@dimensions@tab{#1}%
2540 }

```

`eql@dimensions@addreg`

```

2541 \def\eql@dimensions@addreg#1{#1\the#1\relax}

```

`@dimensions@startrow`

```

2542 \def\eql@dimensions@startrow{%
2543 \eql@dimensions@add{\eql@dimensions@addreg\eql@row@}%
2544 }

```

`@dimensions@savecell`

```

2545 \def\eql@dimensions@savecell{%
2546 \eql@dimensions@add{%
2547 \eql@dimensions@addreg\eql@shape@pos@
2548 \eql@dimensions@addreg\eql@cellwidth@
2549 \eql@dimensions@addreg\eql@shape@amount@
2550 \noexpand\eql@dimensions@cellcall
2551 }%
2552 }

```

`l@dimensions@savesep`

```

2553 \def\eqldimensions@savesep{%
2554   \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2555 }

```

eqldimensions@endrow

```

2556 \def\eqldimensions@endrow{%
2557   \eqldimensions@add{,%
2558     \eqldimensions@addreg\eqldimensions@tagwidth@
2559     \eqldimensions@addreg\eqldimensions@line@height@
2560     \eqldimensions@addreg\eqldimensions@line@depth@
2561     \eqldimensions@addreg\eqldimensions@line@interline@
2562     ;}%
2563 }

```

eqldimensions@saveblocktag

```

2564 \def\eqldimensions@saveblocktag{%
2565   \eqldimensions@add{\eqldimensions@row@0\relax,%
2566     \eqldimensions@tagwidth@block@\the\eqldimensions@tagwidth@\relax
2567     \eqldimensions@tagheight@block@\the\ht\eqldimensions@tagbox@\relax
2568     \eqldimensions@tagdepth@block@\the\dp\eqldimensions@tagbox@\relax
2569     \eqldimensions@addreg\eqldimensions@tagpos@shift@
2570     \let\noexpand\eqldimensions@tagpos@reserve\ifdefined\eqldimensions@tagpos@reserve
2571     \noexpand\eqldimensions@true\else\noexpand\eqldimensions@false\fi
2572     ;}%
2573   \global\eqldimensions@tagwidth@max@\eqldimensions@tagwidth@
2574   \global\eqldimensions@tagrows@\@ne
2575 }

```

eqldimensions@savenoblocktag

```

2576 \def\eqldimensions@savenoblocktag{%
2577   \eqldimensions@add{\eqldimensions@row@0\relax,;%
2578 }

```

\eqldimensions@for

```

2579 \def\eqldimensions@for#1{%
2580   \def\eqldimensions@forcall{#1}%
2581   \expandafter\eqldimensions@forstep\eqldimensions@tab
2582 }

```

eqldimensions@forstep

```

2583 \def\eqldimensions@forstep\eqldimensions@row@#1\relax#2,#3;{%
2584   \eqldimensions@row@#1\relax
2585   \ifnum\eqldimensions@row@=#3\else
2586     #3%
2587     \def\eqldimensions@cells{#2}%
2588     \eqldimensions@forcall
2589     \expandafter\eqldimensions@forstep
2590   \fi
2591 }

```

\eqldimensions@get

```

2592 \def\eqldimensions@get#1{%
2593   \eqldimensions@row@#1\relax

```

```

2594 \expandafter\eqldimensions@getdef\expandafter{\the\eqldimensions@row}%
2595 \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil
2596 }

```

eqldimensions@getdef

```

2597 \def\eqldimensions@getdef#1{%
2598 \def\eqldimensions@getparse
2599   ##1\eqldimensions@row@#1\relax##2,##3;##4\@nil{%
2600   ##3%
2601   \def\eqldimensions@cells{##2}}%
2602 }%
2603 }

```

\eqldimensions@colwidth@tab

```

2604 \let\eqldimensions@colwidth@tab\@empty

```

\eqldimensions@colwidth@get

```

2605 \def\eqldimensions@colwidth@get#1{%
2606 \ifcase\eqldimensions@colwidth@tab\else\z@\fi
2607 }

```

\eqldimensions@colwidth@save

```

2608 \def\eqldimensions@colwidth@save#1{%
2609 \edef\eqldimensions@colwidth@tab{%
2610 \noexpand\or\the#1%
2611 \unexpanded\expandafter{\eqldimensions@colwidth@tab}}%
2612 }%
2613 }

```

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

```

2614 \def\eqldimensions@calc{%
2615 \eqldimensions@column@\z@
2616 \eqldimensions@line@pos@\z@
2617 \eqldimensions@line@possep@\z@
2618 \eqldimensions@line@avail@\eqldimensions@totalwidth@
2619 \eqldimensions@line@availsep@\eqldimensions@intercolumns@
2620 \eqldimensions@line@width@\z@
2621 \eqldimensions@line@widthsep@\z@
2622 \let\eqldimensions@cellcall\eqldimensions@calc@call
2623 \let\eqldimensions@sepcall\eqldimensions@calc@callsep
2624 \eqldimensions@cells
2625 }

```

\eqldimensions@calc@callsep Callback for each intercolumn space.

```

2626 \def\eqldimensions@calc@callsep{%
2627 \advance\eqldimensions@line@possep@\@ne
2628 }%

```

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

```

2629 \def\eqldimensions@calc@call{%
2630   \advance\eqldcolumn@\@ne
2631   \ifnum\eqldtotalcolumns@=\@ne
2632     \dimen@\eqldtotalwidth@
2633   \else
2634     \dimen@\eqldcolwidth@get\eqldcolumn@\relax
2635   \fi
2636   \ifdim\eqldcellwidth@>\z@
2637     \ifdim\eqldline@width@=\z@
2638       \eqldline@avail@\eqldline@pos@
2639       \eqldline@availsep@\eqldline@possep@
2640       \ifcase\eqldshape@pos@
2641         \or
2642         \advance\eqldline@avail@\dimexpr
2643           (\dimen@-\eqldcellwidth@+\eqldcenteroffset@)/\tw@\relax
2644         \or
2645         \advance\eqldline@avail@\dimexpr\dimen@-\eqldcellwidth@\relax
2646       \fi
2647       \advance\eqldline@avail@\eqldshape@amount@
2648     \fi
2649     \eqldline@width@\eqldline@pos@
2650     \eqldline@widthsep@\eqldline@possep@
2651     \ifcase\eqldshape@pos@
2652       \advance\eqldline@width@\eqldcellwidth@
2653     \or
2654       \advance\eqldline@width@\dimexpr
2655         (\dimen@+\eqldcellwidth@+\eqldcenteroffset@)/\tw@\relax
2656     \or
2657       \advance\eqldline@width@\dimen@
2658     \fi
2659     \advance\eqldline@width@\eqldshape@amount@
2660   \fi
2661   \advance\eqldline@pos@\dimen@
2662 }

```

I.4 Best Line Selection

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

```
2663 \let\eqldnumbering@best@auto\eqldfalse
```

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

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

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

```

2664 \newcount\eqldnumbering@best@row@
2665 \newdimen\eqldnumbering@best@space@
2666 \let\eqldnumbering@best@use\eqldfalse

```

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

```

2667 \def\eqldnumbering@best@find{%
2668   \eqldnumbering@best@row@\z@
2669   \eqldnumbering@best@space@\z@
2670   \eqlddimensions@for{%
2671     \eqlddimensions@calc

```

```

2672 \ifdefined\eq\tagleft
2673 \dimen@\eq\line@avail@
2674 \else
2675 \dimen@\dimexpr\eq\totalwidth@-\eq\line@width@\relax
2676 \fi
2677 \ifdim\dimen@>\eq\numbering@best@space@
2678 \eq\numbering@best@row@\eq\row@
2679 \eq\numbering@best@space@\dimen@
2680 \fi
2681 }%
2682 \ifnum\eq\numbering@best@row@>\z@
2683 \eq\tagpos@row@\eq\numbering@best@row@
2684 \let\eq\tagpos@continuous\eq\false
2685 \eq\tagpos@prevrow@\z@
2686 \fi
2687 }

```

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

```

2688 \def\eq\numbering@best@test#1{%
2689 \eq\dimensions@get#1%
2690 \eq\dimensions@calc
2691 \ifdefined\eq\tagleft
2692 \dimen@\dimexpr\eq\line@avail@
2693 +\eq\marginleft@+\eq\line@availsep@\eq\colsep@\relax
2694 \else
2695 \dimen@\dimexpr\displaywidth-\eq\line@width@
2696 -\eq\marginleft@-\eq\line@widthsep@\eq\colsep@\relax
2697 \fi
2698 \ifdim\dimen@<\eq\tagwidth@block@
2699 \let\eq\numbering@best@use\eq>true
2700 \fi
2701 }

```

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

```

2702 \def\eq\numbering@best@eval{%
2703 \ifdefined\eq\numbering@best@auto
2704 \ifdefined\eq\numbering@best@use\else
2705 \ifdefined\eq\numbering@multi\else
2706 \ifnum\eq\tagpos@row@>\z@
2707 \eq\numbering@best@test\eq\tagpos@row@
2708 \fi
2709 \ifnum\eq\tagpos@prevrow@>\z@
2710 \eq\numbering@best@test\eq\tagpos@prevrow@
2711 \fi
2712 \fi
2713 \fi
2714 \fi
2715 \ifdefined\eq\numbering@best@use
2716 \eq\numbering@best@find
2717 \fi
2718 }

```

I.5 Tag Margin

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

djust@calc@tagmargin

```
2719 \def\eql@adjust@calc@tagmargin{%
2720   \ifdefined\eql@tagmargin@val
2721     \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2722   \else
2723     \eql@tagmargin@\eql@tagwidth@max@
2724     \ifdim\eql@tagmargin@>\z@
2725       \advance\eql@tagmargin@-\eql@tagsepmin@
2726     \fi
2727   \fi

2728   \dimen@\eql@tagrows@p@
2729   \ifnum\eql@totalrows@=\@ne
2730     \ifnum\eql@tagrows@=\@ne
2731       \advance\dimen@1sp\relax
2732     \fi
2733   \fi
2734   \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
2735     \eql@tagmargin@\z@
2736   \fi

2737   \@tempdima\dimexpr\displaywidth
2738     -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
2739   \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
2740   \ifdim\@tempdimb>\z@
2741     \ifdim\eql@tagmargin@threshold@\@tempdima<\@tempdimb
2742       \eql@tagmargin@\z@
2743     \fi
2744   \fi
2745 }
```

I.6 Single Column

ql@adjust@calc@lines

```
2746 \def\eql@adjust@calc@lines{%
2747   \eql@totalcolumns@\@ne
2748   \eql@intercolumns@\z@
2749   \eql@colsep@\z@
2750   \ifdefined\eql@layoutleft
2751     \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2752     \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2753     \ifdim\eql@marginleft@<\eql@marginleft@min@
2754       \eql@marginleft@\eql@marginleft@min@
2755     \fi
2756     \dimen@\glueexpr\eql@layoutleftmarginmax\relax
2757     \ifdim\eql@marginleft@>\dimen@
2758       \eql@marginleft@\dimen@
2759     \fi
2760     \eql@marginright@\z@
2761     \eql@centeroffset@\z@
2762   \else
2763     \eql@adjust@calc@tagmargin
2764     \ifdefined\eql@paddingleft@val
2765       \eql@marginleft@\dimexpr
2766         (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2767         -\glueexpr\eql@paddingleft@val\relax\relax
2768       \ifdim\eql@marginleft@<\z@
```



```

2769     \eql@marginleft@z@
2770     \fi
2771   \else
2772     \eql@marginleft@z@
2773     \fi
2774   \ifdefined\eql@paddingright@val
2775     \eql@marginright@dimexpr
2776       (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2777       -\glueexpr\eql@paddingright@val\relax\relax
2778     \ifdim\eql@marginright@<\z@
2779       \eql@marginright@z@
2780     \fi
2781   \else
2782     \eql@marginright@z@
2783     \fi
2784   \ifdim\eql@tagmargin@>z@
2785     \ifdefined\eql@tagsleft
2786       \ifdim\eql@marginleft@<\eql@tagsepmin@
2787         \eql@marginleft@\eql@tagsepmin@
2788         \fi
2789         \advance\eql@marginleft@\eql@tagmargin@
2790         \advance\eql@centeroffset@\eql@tagmargin@
2791       \else
2792         \ifdim\eql@marginright@<\eql@tagsepmin@
2793           \eql@marginright@\eql@tagsepmin@
2794           \fi
2795           \advance\eql@marginright@\eql@tagmargin@
2796           \advance\eql@centeroffset@-\eql@tagmargin@
2797         \fi
2798       \fi
2799     \eql@marginleft@min@z@
2800     \eql@centeroffset@dimexpr\eql@marginright@-\eql@marginleft@
2801     \ifdefined\eql@tagsleft+\else-\fi\eql@tagmargin@\relax
2802   \fi

2803 \eql@totalwidth@dimexpr\displaywidth
2804   -\eql@marginleft@-\eql@marginright@\relax
2805 }

```

I.7 Multiple Columns

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

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

```
2806 \def\eql@adjust@calc@columns{%
```

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

```

2807   \ifnum\eql@totalcolumns@<\thr@@
2808     \eql@totalcolumns@\tw@
2809     \let\eql@columns@fulllength\eql@false
2810   \fi

```

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

```
2811 \eql@intercolumns@numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2812 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

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

```
2813 \ifdefined\eql@layoutleft
2814 \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2815 \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2816 \ifdim\eql@marginleft@<\eql@marginleft@min@
2817 \eql@marginleft@\eql@marginleft@min@
2818 \fi
2819 \else
```

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

```
2820 \eql@adjust@calc@tagmargin
2821 \ifdefined\eql@columns@fulllength
2822 \ifdim\eql@tagmargin@>\z@
2823 \advance\eql@tagmargin@\eql@tagsepmin@
2824 \fi
2825 \fi
2826 \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2827 -\eql@intercolumns@\eql@colsepmin@\relax
2828 \eql@tagmargin@\z@
2829 \fi
2830 \eql@marginleft@min@\z@
2831 \fi
```

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

```
2832 \ifnum\eql@intercolumns@>\z@
```

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

```
2833 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2834 \ifdefined\eql@layoutleft
2835 \advance\eql@colsep@-\eql@marginleft@
2836 \else
2837 \advance\eql@colsep@-\eql@tagmargin@
2838 \fi
2839 \count@\eql@intercolumns@
2840 \ifdefined\eql@columns@fulllength\else
2841 \ifdefined\eql@layoutleft
2842 \advance\count@\@ne
2843 \else
2844 \advance\count@\tw@
2845 \fi
2846 \fi
2847 \divide\eql@colsep@\count@
```

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

```
2848 \ifdim\eql@colsep@<\eql@colsepmin@
```

```

2849     \eql@colsep@\eql@colsepmin@
2850   \else
2851     \ifdefined\eql@columns@fulllength\else
2852       \dimen@\glueexpr\eql@colsepmax@val\relax
2853       \ifdim\eql@colsep@>\dimen@
2854         \eql@colsep@\dimen@
2855       \fi
2856     \fi
2857   \fi
2858 \else

```

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

```

2859     \eql@colsep@\eql@colsepmin@
2860   \fi

```

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

```

2861   \ifdefined\eql@layoutleft

```

Set the default value:

```

2862     \ifdim\eql@colsep@=\eql@colsepmin@

```

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

```

2863     \dimen@\dimexpr\displaywidth-\eql@totalwidth@
2864     -\eql@intercolumns@\eql@colsep@\relax
2865     \ifdim\dimen@<\eql@marginleft@
2866       \ifdim\dimen@<\eql@marginleft@min@
2867         \eql@marginleft@\eql@marginleft@min@
2868       \else
2869         \eql@marginleft@\dimen@
2870       \fi
2871     \fi
2872   \fi
2873 \else

```

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

```

2874     \ifdefined\eql@columns@fulllength
2875       \eql@marginleft@\z@

```

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

```

2876   \else
2877     \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@
2878     -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
2879     \ifdim\eql@marginleft@<\z@
2880       \eql@marginleft@\z@
2881     \fi
2882   \fi

```

Add tag margin in case of left tags:

```

2883     \ifdefined\eql@tagsleft
2884       \advance\eql@marginleft@\eql@tagmargin@
2885     \fi
2886   \fi

```

Find the best row for tag placement:

```
2887 \eql@numbering@best@eval
```

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

```
2888 \ifdefined\eql@tagsleft
2889 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
2890 \else
2891 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2892 \fi
```

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

```
2893 \ifdefined\eql@numbering@multi
2894 \eql@dimensions@for{%
2895 \ifdim\eql@tagwidth@>\z@
2896 \eql@dimensions@calc
2897 \eql@adjust@columns@test
2898 \fi
2899 }%
2900 \else
2901 \ifnum\eql@tagpos@row@>\z@
2902 \ifnum\eql@tagpos@row@>\eql@totalrows@\else
2903 \eql@dimensions@get\eql@tagpos@row@
2904 \eql@tagwidth@\eql@tagwidth@block@
2905 \eql@dimensions@calc
2906 \eql@adjust@columns@test
2907 \fi
2908 \fi
2909 \ifnum\eql@tagpos@prevrow@>\z@
2910 \eql@dimensions@get\eql@tagpos@prevrow@
2911 \eql@tagwidth@\eql@tagwidth@block@
2912 \eql@dimensions@calc
2913 \eql@adjust@columns@test
2914 \fi
2915 \fi
```

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

```
2916 \advance\eql@totalwidth@\dimexpr
2917 \eql@intercolumns@\eql@colsep@+\eql@marginleft@\relax
2918 }
```

Placement for Right Tags.

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

```
2919 \def\eql@adjust@columns@test@tagsright{%
```

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

```
2920 \@tempdima\dimexpr\displaywidth-\eql@linewidth-\eql@tagwidth@\relax
```

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

```
2921 \ifdim\@tempdima<\dimexpr
2922 \eq\marginleft@+\eq\linewidthsep@\eq\colsep@\relax
```

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

```
2923 \ifdim\@tempdima<\dimexpr
2924 \eq\marginleft@min@+\eq\linewidthsep@\eq\colsepmin@\relax\else
```

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

```
2925 \eq\adjust@columns@modify@tagsright
2926 \fi
2927 \fi
2928 }
```

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

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

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

```
2930 \ifnum\eq\linewidthsep@>\z@
2931 \dimen@\@tempdima
2932 \count@\eq\linewidthsep@
2933 \ifdefined\eq\layoutleft
2934 \advance\dimen@-\eq\marginleft@
2935 \else
2936 \ifdefined\eq\columns@fulllength\else
2937 \advance\count@\@ne
2938 \fi
2939 \fi
2940 \divide\dimen@\count@
```

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

```
2941 \ifdim\dimen@<\eq\colsep@
2942 \ifdim\dimen@<\eq\colsepmin@
2943 \eq\colsep@\eq\colsepmin@
2944 \else
2945 \eq\colsep@\dimen@
2946 \fi
2947 \fi
2948 \fi
```

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

```
2949 \dimen@\dimexpr\@tempdima-\eq\linewidthsep@\eq\colsep@\relax
2950 \ifdim\eq\marginleft@>\dimen@
2951 \eq\marginleft@\dimen@
2952 \fi
2953 }
```

Placement for Left Tags.

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

```
2954 \def\eq@adjust@columns@test@tagsleft{%
```

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

```
2955 \count@\numexpr\eq@intercolumns@-\eq@line@availsep@relax
2956 \@tempdima\dimexpr\eq@tagwidth@-\eq@line@avail@relax
```

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

```
2957 \ifdim\@tempdima>\dimexpr
2958 \eq@marginleft@+\eq@line@availsep@relax
```

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

```
2959 \ifdim\eq@tagwidth@<%
2960 \ifdefined\eq@layoutleft
2961 \glueexpr\eq@layoutleftmarginmax@relax
2962 \else
2963 \displaywidth
2964 \fi
```

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

```
2965 \ifdim\@tempdima>\dimexpr
2966 \displaywidth-\eq@totalwidth@-\count@\eq@colsepmin@relax\else
```

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

```
2967 \eq@adjust@columns@modify@tagsleft
2968 \fi
2969 \fi
2970 \fi
2971 }
```

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

```
2972 \def\eq@adjust@columns@modify@tagsleft{%
```

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

```
2973 \ifnum\count@>\z@
2974 \dimen@\dimexpr\displaywidth-\eq@totalwidth@-\@tempdima@relax
2975 \ifdefined\eq@columns@fulllength@else
2976 \advance\count@\@ne
2977 \fi
2978 \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.

```

2979 \ifdim\dimen@<\eql@colsep@
2980 \ifdim\dimen@<\eql@colsepmin@
2981 \dimen@\eql@colsepmin@
2982 \fi
2983 \advance\dimen@-\eql@colsep@
2984 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
2985 \advance\eql@colsep@\dimen@
2986 \fi
2987 \fi

```

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

```

2988 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
2989 \ifdim\eql@marginleft@<\dimen@
2990 \eql@marginleft@\dimen@
2991 \fi
2992 }

```

J Single Column Arrangement

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

J.1 Supporting Definitions

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

```

2993 \ifdefined\inf@bad\else%
2994 \newcount\inf@bad
2995 \inf@bad1000000\relax
2996 \fi

```

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

```

2997 \let\eql@restore@hfuzz\@empty
2998 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}

```

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

```

2999 \newcount\eql@alignbadness@
3000 \newcount\eql@tagbadness@
3001 \newcount\eql@arrange@badness@
3002 \eql@alignbadness@\inf@bad
3003 \eql@tagbadness@\inf@bad

```

J.2 Arrangement Methods

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

```

3004 \def\eql@arrange@try#1{%
3005 \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax

```

```

3006 \setbox\eql@cellbox@\hbox to\displaywidth{%
3007 \unhbox\eql@cellbox@\unkern\kern#1}%
3008 \eql@arrange@badness@\badness
3009 \else
3010 \eql@arrange@badness@\m@ne
3011 \fi
3012 }

```

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

TODO: adjust

```

3013 \def\eql@arrange@print#1#2{%
3014 \eql@restore@hfuzz
3015 \if@eqnsw
3016 \ifdefined\eql@tagsleft
3017 \eql@tagbox@print@tagsleft
3018 \fi
3019 \fi
3020 \hbox to\displaywidth{%
3021 #1%
3022 \unhbox\eql@cellbox@\unkern
3023 #2%
3024 \eql@tagging@mathaddlast
3025 }%
3026 \if@eqnsw
3027 \ifdefined\eql@tagsleft\else
3028 \eql@tagbox@print@tagsright
3029 \fi
3030 \fi
3031 }

```

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

```

3032 \def\eql@arrange@print@alignleft#1#2{%
3033 \eql@display@firstavail@set{\dimexpr#1\relax}%
3034 \eql@arrange@print{\kern#1}{\kern#2}%
3035 }

3036 \def\eql@arrange@print@alignright#1#2{%
3037 \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3038 \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3039 }

3040 \def\eql@arrange@print@aligncenter#1{%
3041 \eql@display@firstavail@set{\dimexpr
3042 (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3043 \ifdim#1>\z@
3044 \eql@arrange@print{\kern#1\hfil}{}%
3045 \else
3046 \eql@arrange@print{\hfil}{\kern-#1}%
3047 \fi
3048 }

```

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

Select the shape scheme:

```

3049 \def\eql@arrange@init{%
3050   \eql@save@hfuzz
3051   \hfuzz\maxdimen
3052   \eql@shape@select
3053 }

```

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

```

3054 \def\eql@arrange@print@line{%
3055   \eql@tagging@tagaddbox
3056   \csname eql@arrange%
3057     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3058     @init\endcsname
3059   \csname eql@arrange%
3060     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3061     @\ifdefined\eql@tagpos@reserve
3062       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3063       notag\fi\endcsname
3064 }

```

J.3 Central Alignment

TODO: describe

```

3065 \def\eql@arrange@aligncenter@init{%
3066   \eql@tagging@aligncenter
3067   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3068     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3069 }

```

TODO: describe

```

3070 \def\eql@arrange@aligncenter@notag{%
3071   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3072     \ifdim\eql@line@offset@<\eql@marginleft@min@
3073       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3074     \else
3075       \eql@line@offset@
3076     \fi
3077   \eql@arrange@print@aligncenter\eql@line@offset@
3078 \else
3079   \ifdim\eql@line@offset@<\eql@marginleft@min@
3080     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3081   \else
3082     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3083   \fi
3084 \fi
3085 }

```

TODO: describe

```

3086 \def\eql@arrange@aligncenter@tagsright{%
3087   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3088     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3089       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3090     \else

```

```

3091     \dimexpr\tw@eql@tagwidth@+\eql@line@offset@relax
3092     \fi
3093     \eql@arrange@print@aligncenter\eql@line@offset@
3094 \else
3095     \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@relax}%
3096     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3097         \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@relax
3098             \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3099         \else
3100             \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3101         \fi
3102     \else
3103         \let\eql@tagpos@reserve\eql@false
3104         \eql@arrange@aligncenter@notag
3105     \fi
3106 \fi
3107 }

3108 \def\eql@arrange@aligncenter@tagsleft{%
3109     \ifdim\eql@tagwidth@>\eql@marginleft@min@
3110         \ifdim\dimexpr\displaywidth-\eql@cellwidth@relax>%
3111             \ifdim\eql@line@offset@<\eql@tagwidth@
3112                 \dimexpr\tw@eql@tagwidth@-\eql@line@offset@relax
3113             \else
3114                 \eql@line@offset@
3115             \fi
3116         \eql@arrange@print@aligncenter\eql@line@offset@
3117     \else
3118         \eql@arrange@try\eql@tagwidth@
3119         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3120             \ifdim\eql@line@offset@<\eql@tagwidth@
3121                 \eql@arrange@print@alignleft\eql@tagwidth@z@
3122             \else
3123                 \eql@arrange@print@alignright\eql@tagwidth@z@
3124             \fi
3125         \else
3126             \let\eql@tagpos@reserve\eql@false
3127             \eql@arrange@aligncenter@notag
3128         \fi
3129     \fi
3130 \else
3131     \eql@arrange@aligncenter@notag
3132 \fi
3133 }

```

J.4 Left Alignment

```

3134 \def\eql@arrange@alignleft@init{%
3135     \eql@tagging@alignleft
3136     \eql@line@offset@\dimexpr\eql@marginleft@+\eql@shape@amount@relax
3137     \ifdim\eql@line@offset@<\eql@marginleft@min@
3138         \eql@line@offset@\eql@marginleft@min@
3139     \fi
3140 }

3141 \def\eql@arrange@alignleft@notag{%
3142     \ifdim\eql@line@offset@>\eql@marginleft@min@
3143         \eql@arrange@try\eql@line@offset@
3144         \ifnum\eql@arrange@badness@<\eql@alignbadness@

```

```

3145     \eql@arrange@print@alignleft\eql@line@offset@\z@
3146     \else
3147     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3148     \fi
3149     \else
3150     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3151     \fi
3152 }

3153 \def\eql@arrange@alignleft@tagsright{%
3154 \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3155 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3156 \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3157 \else
3158 \ifdim\eql@line@offset@>\eql@marginleft@min@
3159 \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3160 \fi
3161 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3162 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3163 \else
3164 \let\eql@tagpos@reserve\eql@false
3165 \eql@arrange@alignleft@notag
3166 \fi
3167 \fi
3168 }

3169 \def\eql@arrange@alignleft@tagsleft{%
3170 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3171 \ifdim\eql@line@offset@>\eql@tagwidth@
3172 \eql@arrange@try\eql@line@offset@
3173 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3174 \eql@arrange@print@alignleft\eql@line@offset@\z@
3175 \else
3176 \eql@arrange@try\eql@tagwidth@
3177 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3178 \eql@arrange@print@alignright\eql@tagwidth@\z@
3179 \else
3180 \let\eql@tagpos@reserve\eql@false
3181 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3182 \fi
3183 \fi
3184 \else
3185 \eql@arrange@try\eql@tagwidth@
3186 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3187 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3188 \else
3189 \let\eql@tagpos@reserve\eql@false
3190 \eql@arrange@alignleft@notag
3191 \fi
3192 \fi
3193 \else
3194 \eql@arrange@alignleft@notag
3195 \fi
3196 }

```

J.5 Right Alignment

```

3197 \def\eql@arrange@alignright@init{%
3198 \eql@tagging@alignright
3199 \eql@line@offset@\dimexpr\eql@marginright@-\eql@shape@amount@\relax

```

```

3200 \ifdim\eq\line@offset@<\z@
3201   \eq\line@offset@\z@
3202 \fi
3203 }

```

TODO: describe

```

3204 \def\eq\arrange@alignright@notag{%
3205   \ifdim\eq\line@offset@>\z@
3206     \eq\arrange@try{\dimexpr\eq\marginleft@min@+\eq\line@offset@\relax}%
3207     \ifnum\eq\arrange@badness@<\eq\alignbadness@
3208       \eq\arrange@print@alignright\eq\marginleft@min@\eq\line@offset@
3209     \else
3210       \eq\arrange@print@alignleft\eq\marginleft@min@\z@
3211     \fi
3212 \else
3213   \eq\arrange@print@alignright\eq\marginleft@min@\z@
3214 \fi
3215 }

```

TODO: describe

```

3216 \def\eq\arrange@alignright@tagsright{%
3217   \ifdim\eq\line@offset@>\eq\tagwidth@
3218     \eq\arrange@try{\dimexpr\eq\marginleft@min@+\eq\line@offset@\relax}%
3219     \ifnum\eq\arrange@badness@<\eq\alignbadness@
3220       \eq\arrange@print@alignright\eq\marginleft@min@\eq\line@offset@
3221     \else
3222       \eq\arrange@try{\dimexpr\eq\marginleft@min@+\eq\tagwidth@\relax}%
3223       \ifnum\eq\arrange@badness@<\eq\tagbadness@
3224         \eq\arrange@print@alignleft\eq\marginleft@min@\eq\tagwidth@
3225       \else
3226         \let\eq\tagpos@reserve\eq>false
3227         \eq\arrange@print@alignleft\eq\marginleft@min@\z@
3228       \fi
3229     \fi
3230 \else
3231   \eq\arrange@try{\dimexpr\eq\marginleft@min@+\eq\tagwidth@\relax}%
3232   \ifnum\eq\arrange@badness@<\eq\tagbadness@
3233     \eq\arrange@print@alignright\eq\marginleft@min@\eq\tagwidth@
3234   \else
3235     \let\eq\tagpos@reserve\eq>false
3236     \eq\arrange@alignright@notag
3237   \fi
3238 \fi
3239 }

```

TODO: describe

```

3240 \def\eq\arrange@alignright@tagsleft{%
3241   \ifdim\eq\tagwidth@>\eq\marginleft@min@
3242     \eq\arrange@try{\dimexpr\eq\line@offset@+\eq\tagwidth@\relax}%
3243     \ifnum\eq\arrange@badness@<\eq\alignbadness@
3244       \eq\arrange@print@alignright\eq\tagwidth@\eq\line@offset@
3245     \else
3246       \ifdim\eq\line@offset@>\z@
3247         \eq\arrange@try\eq\tagwidth@
3248       \fi
3249       \ifnum\eq\arrange@badness@<\eq\tagbadness@
3250         \eq\arrange@print@alignleft\eq\tagwidth@\z@
3251       \else

```

```

3252     \let\eql@tagpos@reserve\eql@false
3253     \eql@arrange@alignright@notag
3254     \fi
3255     \fi
3256   \else
3257     \eql@arrange@alignright@notag
3258     \fi
3259 }

```

K Equations Box Environment

TODO: outline sequence of calls

TODO: describe

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

TODO: vspace?!

K.1 Line Breaks

`\eql@box@cr`

```

3260 \protected\def\eql@box@cr{%
3261   \eql@ampprotecttwo{\eql@ifnextchar@tight[]\eql@box@cr@skip\eql@box@cr@
3262 }
3263 \def\eql@box@cr@f{%
3264   \eql@punct@apply@line
3265   \eql@hook@lineout
3266   \eql@box@lastcell
3267   \cr
3268 }
3269 \def\eql@box@cr@skip[#1]{%
3270   \eql@box@cr@
3271   \noalign{%
3272     \vskip\glueexpr#1\relax
3273   }%
3274 }

```

K.2 Stacked Mode

```

3275 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3276 \def\eql@box@open@stacked{%
3277   \eql@shape@align@enable
3278   \let\eql@box@lastcell\eql@box@lastcell@stacked
3279   \everycr{\noalign{%
3280 (dev)\eql@dev{starting line \the\eql@row}}%
3281     \global\advance\eql@row@\@ne
3282   }}%
3283   \tabskip\z@skip
3284   \halign\bgroup
3285     &%
3286     \global\let\eql@cell@container\@empty
3287     \setbox\eql@cellbox@\hbox{%
3288       \eql@strut@cell
3289       \lign

```

```

3290     $\m@th\displaystyle
3291     \eql@hook@colin
3292     ##%
3293     \eql@punct@apply@col
3294     \eql@hook@colout
3295     \eql@tagging@mathsave
3296     $%
3297     \eql@tagging@mathaddlast
3298     }%
3299     \ifdefined\eql@shape@lastrow
3300     \eql@totalrows@\eql@row@
3301     \fi
3302     \eql@shape@eval
3303     \eql@cell@container
3304     \ifdefined\eql@frame@cmd
3305     \ifcase\eql@shape@pos@
3306     \eql@frame@measure
3307     \advance\eql@shape@amount@-\eql@frame@margin@
3308     \or\or
3309     \eql@frame@measure
3310     \advance\eql@shape@amount@+\eql@frame@margin@
3311     \fi
3312     \eql@frame@print
3313     \fi
3314     \ifcase\eql@shape@pos@
3315     \kern\eql@shape@amount@
3316     \box\eql@cellbox@
3317     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3318     -\eql@shape@amount@+\@flushglue\relax
3319     \eql@tagging@alignleft
3320     \or
3321     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3322     \box\eql@cellbox@
3323     \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3324     \eql@tagging@aligncenter
3325     \or
3326     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3327     +\eql@shape@amount@+\@flushglue\relax
3328     \box\eql@cellbox@
3329     \kern-\eql@shape@amount@
3330     \eql@tagging@alignright
3331     \fi
3332     \tabskip\eql@colsep@\relax
3333     \crrc
3334     \noalign{%
3335     \global\let\eql@shape@lastrow\eql@false
3336     \eql@hook@blockbefore
3337     }%
3338     \eql@hook@blockin
3339 }

3340 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

K.3 Aligned Mode

```

3341 \def\eql@box@lastcell@odd{%
3342 &\omit
3343 \eql@prevwidth@\wd\eql@cellbox@
3344 \let\eql@frame@cmd\eql@frame@prevcmd

```

```

3345 \ifdefined\eql@frame@cmd
3346   \eql@frame@measure
3347   \advance\eql@prevwidth@\eql@frame@margin@
3348   \eql@frame@print
3349   \fi
3350   \kern-\eql@prevwidth@
3351   \unhbox\eql@cellbox@
3352   \hfil
3353   &\omit\kern-\eql@colsep@
3354 }%
3355 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}

3356 \def\eql@box@open@aligned{%
3357 % \TODO templates
3358   \eql@shape@align@disable
3359   \let\eql@box@lastcell\@empty
3360   \everycr{\noalign{%
3361 (dev)\eql@dev{starting new line}%
3362 }}%
3363   \tabskip\z@skip
3364   \halign\bgroup
3365     &%
3366     \let\eql@box@lastcell\eql@box@lastcell@odd
3367     \global\let\eql@cell@container\@empty
3368     \global\setbox\eql@cellbox@\hbox{%
3369       \eql@strut@cell
3370       \@lign
3371       $\m@th\displaystyle
3372       \eql@hook@colin
3373       ##%
3374       \eql@class@innerleft
3375       \eql@hook@innerleft
3376       \eql@tagging@mathsave
3377       $%
3378       \eql@tagging@mathaddlast
3379     }%
3380     \eql@cell@container
3381     \hfil
3382     \kern\wd\eql@cellbox@
3383     \ifdefined\eql@frame@cmd
3384       \eql@frame@measure
3385       \kern\eql@frame@margin@
3386     \fi
3387     \global\let\eql@frame@prevcmd\eql@frame@cmd
3388     \tabskip\z@skip
3389     &%
3390     \eql@prevwidth@\wd\eql@cellbox@
3391     \let\eql@box@lastcell\eql@box@lastcell@even
3392     \let\eql@frame@cmd\eql@frame@prevcmd
3393     \global\let\eql@cell@container\@empty
3394     \setbox\eql@cellbox@\hbox{%
3395       \unhbox\eql@cellbox@
3396       \eql@strut@cell
3397       \@lign
3398       $\m@th\displaystyle
3399       \eql@hook@innerright
3400       \eql@class@innerright@sel
3401       ##%
3402       \eql@punct@apply@col

```

```

3403     \eql@hook@colout
3404     \eql@tagging@mathsave
3405     $%
3406     \eql@tagging@mathaddlast
3407   }%
3408   \eql@cell@container
3409   \ifdefined\eql@frame@cmd
3410     \eql@frame@measure
3411     \advance\eql@prevwidth@\eql@frame@margin@
3412     \eql@frame@print
3413   \fi
3414   \kern-\eql@prevwidth@
3415   \unhbox\eql@cellbox@
3416   \hfil
3417   \tabskip\eql@colsep@\relax
3418 \crrr
3419 \noalign{%
3420   \eql@hook@blockbefore
3421 }%
3422 \eql@hook@blockin
3423 }
3424 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

K.4 Main

```

3425 \let\eql@box@box\vcenter
3426 \let\eql@box@open\@undefined
3427 \let\eql@box@frame\@firstofone
3428 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{##1#2}}

```

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

```

3429 \def\eql@box@close{%
3430   \ifvmode\else
3431     \global\let\eql@shape@lastrow\eql@true
3432     \eql@punct@apply@block
3433     \eql@box@cr@
3434   \fi
3435   \noalign{%
3436     \eql@hook@blockafter
3437     \global\let\eql@shape@lastrow\eql@false
3438   }%
3439   \eql@tagging@tablesaveinner
3440 \egroup
3441 }

```

`\eql@box@vcenter`

```

3442 \def\eql@box@vcenter#1{%
3443   \ifmmode
3444     \vcenter{#1}%
3445   \else
3446     $\m@th\vcenter{#1}$%
3447   \fi
3448 }

```


`\eql@box@start`

```
3449 \let\eql@box@endmath\eql@false
3450 \def\eql@box@start{%
3451   \relax
3452   \ifmmode
3453     \let\eql@box@endmath\eql@false
3454   \else
3455     \let\eql@box@endmath\eql@true
3456     \expandafter$%$
3457   \fi
3458   \eql@box@processopt
3459   \eql@stack@save@box
3460   \let\eql@frame@cmd\@undefined
3461   \let\eql@layoutleft\eql@false
3462   \eql@row@z@
3463   \eql@totalrows@\@M
3464   \eql@shape@select
3465   \setbox\z@\ifx\eql@box@box\vcenter
3466     \expandafter\vbox
3467   \else
3468     \expandafter\eql@box@box
3469   \fi\bgroup
3470   \eql@display@nest
3471   \let\\\eql@box@cr
3472   \eql@spread@set
3473   \eql@strut@make
3474   \eql@box@open
3475 }
```

`\eql@box@end`

```
3476 \def\eql@box@end{%
3477   \eql@box@close
3478   \egroup
3479   \eql@box@frame{%
3480     \ifdefined\eql@display@marginleft
3481       \hskip\glueexpr\eql@display@marginleft\relax
3482     \fi
3483     \ifx\eql@box@box\vcenter
3484       \eql@box@vcenter{\unvbox\z@}%
3485     \else
3486       \box\z@
3487     \fi
3488     \eql@tagging@tableaddinner
3489     \ifdefined\eql@display@marginright
3490       \hskip\glueexpr\eql@display@marginright\relax
3491     \fi
3492   }%
3493   \eql@stack@restore
3494   \ifdefined\eql@box@endmath
3495     \expandafter$%$
3496   \fi
3497 }
```

K.5 Environment

`equationsbox` (*env.*)

```
3498 \newenvironment{equationsbox}{%
3499 (dev)\eql@dev@enterenv
3500 \eql@ampprotect\eql@box@testall\eql@box@start
3501 }{%
3502 \eql@box@end
3503 (dev)\eql@dev@leaveenv
3504 }

3505 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3506 \def\eql@box@parseopt{%
3507 \ifx\eql@parseopt@token[%]
3508 \let\eql@parseopt@next\eql@parseopt@opt
3509 \fi
3510 \ifx\eql@parseopt@token=%
3511 \let\eql@parseopt@next\eql@parseopt@lines
3512 \fi
3513 \ifx\eql@parseopt@token|
3514 \let\eql@parseopt@next\eql@parseopt@columns
3515 \fi
3516 }
```

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

```
3517 \def\eql@box@processopt{%
3518 \let\eql@box@frame\@firstofone
3519 \let\eql@display@marginleft\@undefined
3520 \let\eql@display@marginright\@undefined
3521 \eql@nextopt@process{equationsbox}%
3522 \let\eql@punct@block\eql@punct@main
3523 \let\eql@punct@main\relax
3524 \eql@colsep@\glueexpr\eql@box@colsep\relax
3525 \ifdefined\eql@paddingleft@val
3526 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3527 \else
3528 \eql@paddingleft@\z@
3529 \fi
3530 \ifdefined\eql@paddingright@val
3531 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3532 \else
3533 \eql@paddingright@\z@
3534 \fi
3535 \eql@indent@\glueexpr\eql@indent@val\relax
3536 }
```

L Single-Line Equation

TODO: describe

L.1 Native Mode

```
3537 \def\eql@single@start@native{%
3538 \eql@display@init
3539 \eql@display@print
```

```

3540 \let\raisetag\eqL@raisetag@default
3541 \eqL@shape@align@disable
3542 \eqL@hook@eqin
3543 % \mathopen{ }%
3544 }%

```

TODO: describe

```

3545 \def\eqL@single@end@native{%
3546 % \mathclose{ }%
3547 \eqL@tags@container
3548 \eqL@numbering@single@eval
3549 \if@eqnsw
3550 \ifdefined\eqL@tagsleft
3551 \leqno
3552 \else
3553 \eqno
3554 \fi
3555 \eqL@composetag@print
3556 \fi
3557 \eqL@interline@container
3558 \advance\eqL@belowspace@\eqL@vspaceskip@
3559 \eqL@display@container
3560 \eqL@display@penalty
3561 \eqL@display@vspace@native
3562 }%

```

L.2 Print

```

3563 \def\eqL@single@start@print{%
3564 \eqL@display@init
3565 \eqL@display@print
3566 \eqL@shape@align@enable
3567 \eqL@totalrows@\@ne
3568 \eqL@row@\@ne
3569 \eqL@arrange@init
3570 \global\let\eqL@cell@container\@empty
3571 \prevgraf\numexpr\prevgraf+\@ne\relax
3572 \setbox\eqL@cellbox@\hbox\bgroup
3573 \eqL@restore@hfuzz
3574 \eqL@strut@cell
3575 $\m@th\displaystyle%$
3576 \eqL@hook@eqin
3577 }
3578 \def\eqL@single@end@print{%
3579 \eqL@tagging@mathsave
3580 $$
3581 \hfil
3582 \kern\z@
3583 \egroup
3584 \prevgraf\numexpr\prevgraf-\@ne\relax
3585 \eqL@shape@eval
3586 \eqL@cell@container
3587 \ifdefined\eqL@frame@cmd
3588 \eqL@frame@adjust
3589 \fi
3590 \eqL@cellwidth@\wd\eqL@cellbox@

```

```

3591 \eql@line@height@\ht\eql@cellbox@
3592 \eql@line@depth@\dp\eql@cellbox@
3593 \eql@totalwidth@\eql@cellwidth@
3594 \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
3595 \eql@topheight@\eql@line@height@
3596 \eql@bottomdepth@\eql@line@depth@

3597 \eql@tags@container
3598 \eql@numbering@single@eval
3599 \if@eqnsw
3600 \eql@tagbox@make\eql@composetag@print
3601 \eql@tagrows@\@ne
3602 \ifdefined\eql@tagpos@reserve\else
3603 \eql@tagwidth@\z@
3604 \fi
3605 \eql@tagheight@block@\ht\eql@tagbox@
3606 \eql@tagdepth@block@\dp\eql@tagbox@
3607 \else
3608 \eql@numbering@warnunused
3609 \eql@tagwidth@\z@
3610 \eql@tagrows@\z@
3611 \fi
3612 \eql@tagwidth@max@\eql@tagwidth@
3613 \eql@tagpos@single@eval
3614 \eql@tagpos@print@line@eval

3615 \eql@intercolumns@\z@
3616 \eql@adjust@calc@lines

3617 \eql@display@halign@init{ }%
3618 \halign{##\cr cr
3619 \noalign{\eql@display@halign@start}%
3620 \eql@arrange@print@line
3621 \cr
3622 \noalign{\eql@display@halign@end}%
3623 \eql@tagging@tablesavelines
3624 }%
3625 \eql@tagpos@print@line@end
3626 \eql@display@close
3627 }

```

M Multi-Line with Single Column

TODO: outline sequence of calls

M.1 Measure

TODO: describe

```

3628 \def\eql@lines@measure@line@begin{%
3629 (dev)\eql@dev{starting line \the\eql@row@}%
3630 \eql@numbering@measure@line@begin
3631 \eql@hook@linein
3632 }

```

TODO: describe

```

3633 \def\eql@lines@measure@line@end{%
3634 \eql@punct@apply@line

```

```

3635 \eql@hook@lineout
3636 }

```

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

```

3637 \def\eql@lines@measure@cell{%
3638 \ifdefined\eql@frame@cmd
3639 \ifcase\eql@shape@pos@
3640 \eql@frame@measure
3641 \advance\eql@shape@amount@-\eql@frame@margin@
3642 \or\or
3643 \eql@frame@measure
3644 \advance\eql@shape@amount@+\eql@frame@margin@
3645 \fi
3646 \eql@frame@print
3647 \fi
3648 \eql@cellwidth@\wd\eql@cellbox@
3649 \eql@line@height@\ht\eql@cellbox@
3650 \eql@line@depth@\dp\eql@cellbox@
3651 \eql@dimensions@startrow
3652 \eql@dimensions@savecell
3653 \kern\eql@cellwidth@
3654 }

```

`\eql@lines@measure`

```

3655 \def\eql@lines@measure{%
3656 <dev>\eql@dev@enter\eql@lines@measure
3657 \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3658 \eql@totalrows@\@M
3659 \eql@shape@select

3660 \setbox\z@\vbox{\measuring@true\halign{%
3661 \global\let\eql@cell@container\@empty
3662 \setbox\eql@cellbox@\hbox{%
3663 \eql@strut@cell
3664 \@lign
3665 $\m@th\displaystyle
3666 \eql@hook@colin
3667 ##%
3668 \eql@punct@apply@col
3669 \eql@hook@colout
3670 $%
3671 }%
3672 \ifdefined\eql@shape@lastrow
3673 \eql@totalrows@\eql@row@
3674 \fi
3675 \eql@shape@eval
3676 \eql@cell@container
3677 \eql@lines@measure@cell
3678 \eql@measure@tag
3679 \eql@measure@endrow
3680 \crrc

3681 \noalign{%
3682 \global\let\eql@shape@lastrow\eql@false
3683 \eql@hook@blockbefore
3684 }%
3685 \eql@hook@blockin

```

```

3686 \eql@scan@body
3687 \ifvmode\else
3688 \global\let\eql@shape@lastrow\eql@true
3689 \eql@punct@apply@block
3690 \eql@hook@blockout
3691 \eql@display@endline
3692 \cr
3693 \fi
3694 \omit
3695 \cr
3696 \noalign{%
3697 \eql@hook@blockafter
3698 \global\let\eql@shape@lastrow\eql@false
3699 }%
3700 }%

3701 \eql@measure@close

3702 \setbox\z@\vbox{%
3703 \unvbox\z@
3704 \unpenalty
3705 \global\setbox\@ne\lastbox
3706 }%
3707 \eql@totalwidth@\wd\@ne

3708 (dev)\eql@dev@leave\eql@lines@measure
3709 }

```

M.2 Column Placement

TODO: describe Find the best row for tag placement:

```

3710 \def\eql@lines@adjust{%
3711 \eql@tagpos@adjust@eval
3712 \eql@adjust@calc@lines
3713 \eql@numbering@best@eval
3714 }

```

M.3 Print

TODO: describe

mes@print@line@begin

```

3715 \def\eql@lines@print@line@begin{%
3716 (dev)\eql@dev{starting line \the\eql@row}%
3717 \eql@numbering@print@line@begin
3718 \eql@hook@linein
3719 }

```

TODO: describe

```

3720 \def\eql@lines@print@line@end{%
3721 \eql@punct@apply@line
3722 \eql@hook@lineout
3723 }

```

TODO: describe

```

3724 \def\eq@lines@print@line@adjust{%
3725   \ifdefined\eq@frame@cmd
3726     \ifcase\eq@shape@pos@
3727       \eq@frame@measure
3728       \advance\eq@shape@amount@-\eq@frame@margin@
3729     \or\or
3730       \eq@frame@measure
3731       \advance\eq@shape@amount@+\eq@frame@margin@
3732     \fi
3733   \eq@frame@adjust
3734 \fi
3735 \eq@cellwidth@\wd\eq@cellbox@
3736 \eq@line@height@\ht\eq@cellbox@
3737 \eq@line@depth@\dp\eq@cellbox@
3738 \eq@numbering@print@line@eval
3739 \if@eqnsw
3740   \eq@tagbox@make\eq@composetag@print
3741 \fi
3742 \eq@tagpos@print@line@eval
3743 \eq@arrange@print@line
3744 \eq@tagpos@print@line@end
3745 }

```

TODO: describe

```

3746 \def\eq@lines@print{%
3747 (dev)\eq@dev@enter\eq@lines@print
3748 \eq@arrange@init
3749 \eq@display@halign@init\eq@lines@print@line@begin
3750 \eq@display@halign@letcr\eq@lines@print@line@end
3751 \tabskip\z@skip

3752 \halign{%
3753   \global\let\eq@cell@container\@empty
3754   \setbox\eq@cellbox@\hbox{%
3755     \eq@restore@hfuzz
3756     \eq@strut@cell
3757     \@lign
3758     $\m@th\displaystyle
3759     \eq@hook@colin
3760     ##%
3761     \eq@punct@apply@col
3762     \eq@hook@colout
3763     \eq@tagging@mathsave
3764     $%
3765     \hfil
3766     \kern\z@
3767   }%
3768   \eq@shape@eval
3769   \eq@cell@container
3770   \eq@lines@print@line@adjust
3771 \crr

3772 \noalign{%
3773   \eq@display@halign@start
3774   \eq@numbering@print@block@begin
3775   \eq@hook@blockbefore
3776 }%
3777 \eq@hook@blockin
3778 \eq@scan@body

```

```

3779 \ifvmode\else
3780 \relax
3781 \eql@punct@apply@block
3782 \eql@hook@blockout
3783 \eql@display@endline
3784 \cr
3785 \fi
3786 \noalign{%
3787 \eql@hook@blockafter
3788 \eql@display@halign@end
3789 (dev)\eql@dev@leave\eql@lines@print
3790 }%
3791 \eql@tagging@tablesavelines
3792 }%
3793 }

```

N Multi-Line with Multiple Columns

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

N.1 Support

TODO: describe

```

\eql@columns@add@amp
@columns@completerow

```

```

3794 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3795 \def\eql@columns@completerow{%
3796 \count@numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
3797 \edef\eql@tmp{%
3798 \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3799 \eql@tmp
3800 }

3801 \def\eql@columns@overfull{%
3802 \dimen@eql@line@width@
3803 \advance\dimen@-\hfuzz
3804 \ifdim\dimen@>\displaywidth
3805 \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3806 \wd\z@\z@
3807 \ht\z@\eql@line@height@
3808 \dp\z@\eql@line@depth@
3809 \box\z@
3810 \fi
3811 }

```

N.2 Transpose

TODO: describe

TODO: describe

```

3812 \let\eql@transpose@active\eql@false
3813 \def\eql@transpose@end{\eql@transpose@end}
3814 \def\eql@transpose@skip{&\eqnpunct{}}
3815 \def\eql@transpose@complete{%

```



```
3816 \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}
```

TODO: describe

```
3817 \def\eql@transpose{%
3818 \eql@totalcolumns@\z@
3819 \eql@totalrows@\z@
3820 \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
3821 \eql@scan@reg@{}}%
3822 \eql@row@\z@
3823 \eql@transpose@output@row
3824 }
```

TODO: describe

```
3825 \def\eql@transpose@save@col#1{%
3826 \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
3827 \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}
```

TODO: describe

```
3828 \def\eql@transpose@scan@col#1\&{%
3829 \def\@tempa{#1}%
3830 \ifx\@tempa\eql@transpose@end\else
3831 \advance\eql@totalcolumns@\@ne
3832 \eql@row@\z@
3833 \let\eql@transpose@data@col\@empty
3834 \eql@transpose@scan@row#1\\eql@transpose@end\\%
3835 \ifnum\eql@row@>\eql@totalrows@
3836 \eql@totalrows@\eql@row@
3837 \fi
3838 \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
3839 \expandafter\eql@transpose@scan@col
3840 \fi
3841 }
```

TODO: describe

```
3842 \def\eql@transpose@append@row#1{%
3843 \advance\eql@row@\@ne
3844 \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}
```

TODO: describe

```
3845 \def\eql@transpose@scan@row#1\\{%
3846 \def\@tempa{#1}%
3847 \ifx\@tempa\eql@transpose@end\else
3848 \ifx\eql@transpose@active+
3849 \eql@transpose@scan@cell#1&\eql@transpose@end&%
3850 \else
3851 \eql@transpose@append@row{#1}%
3852 \fi
3853 \expandafter\eql@transpose@scan@row
3854 \fi
3855 }
```

TODO: describe

```
3856 \def\eql@transpose@scan@cell#1&#2&{%
3857 \def\@tempa{#2}%
3858 \ifx\@tempa\eql@transpose@end
3859 \eql@transpose@append@row{#1}%
```

```

3860 \else
3861   \eql@transpose@append@row{#1&#2}%
3862   \expandafter\eql@transpose@scan@cell@next
3863 \fi
3864 }

```

TODO: describe

```

3865 \def\eql@transpose@scan@cell@next#1&{%
3866   \def\@tempa{#1}%
3867   \ifx\@tempa\eql@transpose@end\else
3868     \eql@transpose@append@row{&#1}%
3869     \expandafter\eql@transpose@scan@cell@next
3870   \fi
3871 }

```

TODO: describe

```

3872 \def\eql@transpose@output@row{%
3873   \ifnum\eql@row@<\eql@totalrows@
3874     \advance\eql@row@\@ne
3875     \eql@column@\z@
3876     \eql@transpose@output@col
3877     \ifnum\eql@row@<\eql@totalrows@
3878       \eql@scan@addto\\%
3879     \fi
3880     \expandafter\eql@transpose@output@row
3881   \fi
3882 }

```

TODO: describe

```

3883 \def\eql@transpose@output@col{%
3884   \ifnum\eql@column@<\eql@totalcolumns@
3885     \advance\eql@column@\@ne
3886     \csname eql@transpose@data@col@\the\eql@column@\endcsname
3887     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
3888     \ifnum\eql@column@<\eql@totalcolumns@
3889       \eql@scan@addto{\eql@transpose@complete}%
3890     \fi
3891     \expandafter\eql@transpose@output@col
3892   \fi
3893 }

```

N.3 Measure

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

s@measure@line@begin

```

3894 \def\eql@columns@measure@line@begin{%
3895 (dev)\eql@dev{starting line \the\eql@row@}%
3896   \global\eql@column@\z@
3897   \global\eql@line@height@\z@
3898   \global\eql@line@depth@\z@
3899   \eql@numbering@measure@line@begin
3900   \eql@hook@linein
3901 }

```

```

3902 \def\eql@columns@measure@cell{%

```

```

3903 \eq@cellwidth@\wd\eq@cellbox@
3904 \ifdefined\eq@frame@cmd
3905   \eq@frame@measure
3906   \advance\eq@cellwidth@\eq@frame@margin@
3907 \fi
3908 \ifdim\ht\eq@cellbox@>\eq@line@height@
3909   \global\eq@line@height@\ht\eq@cellbox@
3910 \fi
3911 \ifdim\dp\eq@cellbox@>\eq@line@depth@
3912   \global\eq@line@depth@\dp\eq@cellbox@
3913 \fi
3914 \ifnum\eq@column@=\@ne
3915   \eq@dimensions@startrow
3916 \fi
3917 \ifodd\eq@column@
3918   \eq@shape@pos@\tw@
3919 \else
3920   \eq@shape@pos@\z@
3921 \fi
3922 \eq@shape@amount@\z@
3923 \eq@dimensions@savercell
3924 \ifodd\eq@column@\else
3925   \eq@dimensions@savesep
3926 \fi
3927 \kern\eq@cellwidth@
3928 }

```

mns@measure@line@end

```

3929 \def\eq@columns@measure@line@end{%
3930   \eq@punct@apply@line
3931   \eq@hook@lineout
3932   &\omit
3933   \ifnum\eq@column@>\eq@totalcolumns@
3934     \global\eq@totalcolumns@\eq@column@
3935   \fi

```

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

```

3936 \ifdefined\eq@frame@cmd
3937   \advance\eq@column@\@ne
3938   \wd\eq@cellbox@\z@
3939   \eq@columns@measure@cell
3940 \fi
3941 \eq@measure@tag
3942 \eq@measure@endrow
3943 }

```

\eq@columns@measure

```

3944 \def\eq@columns@measure{%
3945 <dev>\eq@dev@center\eq@columns@measure
3946   \eq@totalcolumns@\z@
3947   \eq@measure@init\eq@columns@measure@line@begin\eq@columns@measure@line@end

3948   \setbox\z@\vbox{\measuring@true\halign{%
3949     &%

```

```

3950 \global\advance\eql@column@\@ne
3951 \global\let\eql@cell@container\@empty
3952 \global\setbox\eql@cellbox@\hbox{%
3953 \eql@strut@cell
3954 \@lign
3955 $\m@th\displaystyle
3956 \eql@hook@colin
3957 ##%
3958 \eql@class@innerleft
3959 \eql@hook@innerleft
3960 $%
3961 }%
3962 \eql@cell@container
3963 \hfil
3964 \eql@columns@measure@cell
3965 \global\let\eql@frame@prevcmd\eql@frame@cmd
3966 &%
3967 \eql@prevwidth@\wd\eql@cellbox@
3968 \let\eql@frame@cmd\eql@frame@prevcmd
3969 \global\advance\eql@column@\@ne
3970 \global\let\eql@cell@container\@empty
3971 \setbox\eql@cellbox@\hbox{%
3972 \eql@strut@cell
3973 \@lign
3974 $\m@th\displaystyle
3975 \eql@hook@innerright
3976 \eql@class@innerright@sel
3977 ##%
3978 \eql@punct@apply@col
3979 \eql@hook@colout
3980 $%
3981 }%
3982 \eql@cell@container
3983 \eql@columns@measure@cell
3984 \hfil
3985 \crcr

3986 \noalign{%
3987 \eql@hook@blockbefore
3988 }%
3989 \eql@hook@blockin
3990 \eql@scan@body

3991 \ifvmode\else
3992 \eql@punct@apply@block
3993 \eql@hook@blockout
3994 \eql@display@endline
3995 \cr
3996 \fi
3997 \noalign{%
3998 \eql@hook@blockafter
3999 }%

4000 \omit
4001 \eql@column@\@ne
4002 \eql@columns@completerow
4003 \cr
4004 }}%

```

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

```

4005 \eql@measure@close

4006 \setbox\z@\vbox{%
4007   \unvbox\z@
4008   \unpenalty
4009   \global\setbox\@ne\lastbox
4010 }%
4011 \eql@totalwidth@\wd\@ne

TODO: why not recycle box contents altogether?!

4012 \let\eql@colwidth@tab\@empty
4013 \loop
4014   \setbox\@ne\hbox{%
4015     \unhbox\@ne
4016     \unskip
4017     \global\setbox\thr@\@lastbox
4018   }%
4019 \ifhbox\thr@@
4020   \eql@colwidth@save{\wd\thr@@}%
4021 \repeat

4022 (dev)\eql@dev@leave\eql@columns@measure
4023 }

```

N.4 Columns Placement

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

```

4024 \def\eql@columns@adjust{%
4025   \ifodd\eql@totalcolumns@
4026     \advance\eql@totalcolumns@\@ne
4027   \fi
4028   \eql@tagpos@adjust@eval
4029   \eql@adjust@calc@columns
4030 }

```

N.5 Print

TODO: describe

mns@print@line@begin

```

4031 \def\eql@columns@print@line@begin{%
4032 (dev)\eql@dev{starting line \the\eql@row}%
4033 \global\eql@column@\z@
4034 \global\eql@line@pos@\eql@marginleft@
4035 \global\eql@line@width@\z@
4036 \global\eql@line@avail@\eql@totalwidth@
4037 \global\eql@line@height@\z@
4038 \global\eql@line@depth@\z@
4039 \eql@numbering@print@line@begin
4040 \eql@hook@linein
4041 }

```

l@columns@print@cell

```

4042 \def\eql@columns@print@cell{%
4043   \eql@cellwidth@\wd\eql@cellbox@
4044   \ifodd\eql@column@
4045     \ifdefined\eql@frame@cmd
4046       \eql@frame@measure
4047       \advance\eql@cellwidth@\eql@frame@margin@
4048     \fi
4049     \dimen@z@
4050   \else
4051     \advance\eql@cellwidth@-\eql@prevwidth@

draw a frame

4052   \ifdefined\eql@frame@cmd
4053     \eql@frame@measure
4054     \advance\eql@cellwidth@\eql@frame@margin@
4055     \advance\eql@prevwidth@\eql@frame@margin@
4056     \eql@frame@print
4057   \fi

update height and depth

4058   \ifdim\ht\eql@cellbox@>\eql@line@height@
4059     \global\eql@line@height@\ht\eql@cellbox@
4060   \fi
4061   \ifdim\dp\eql@cellbox@>\eql@line@depth@
4062     \global\eql@line@depth@\dp\eql@cellbox@
4063   \fi

print box

4064   \kern-\eql@prevwidth@
4065   \unhbox\eql@cellbox@
4066   \dimen@-\eql@cellwidth@
4067   \fi

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

4068   \advance\dimen@\eql@colwidth@get\eql@column@\relax
4069   \kern\dimen@

update available and used space

4070   \dimen@\eql@colwidth@get\eql@column@\relax
4071   \ifdim\eql@cellwidth@>z@
4072     \ifdim\eql@line@width@=z@
4073       \eql@line@avail@\eql@line@pos@
4074       \ifodd\eql@column@
4075         \advance\eql@line@avail@\dimen@
4076         \advance\eql@line@avail@-\eql@cellwidth@
4077       \fi
4078       \global\eql@line@avail@\eql@line@avail@
4079     \fi
4080     \eql@line@width@\eql@line@pos@
4081     \ifodd\eql@column@
4082       \advance\eql@line@width@\dimen@
4083     \else
4084       \advance\eql@line@width@\eql@cellwidth@
4085     \fi
4086     \global\eql@line@width@\eql@line@width@
4087   \fi

```

```

4088 \advance\eqL@line@pos@\dimen@
4089 \ifodd\eqL@column@\else
4090   \advance\eqL@line@pos@\eqL@colsep@
4091 \fi
4092 \global\eqL@line@pos@\eqL@line@pos@
4093 }

```

```

4094 \def\eqL@columns@print@trailright{%
4095   &\omit
4096   \eqL@prevwidth@\wd\eqL@cellbox@
4097   \let\eqL@frame@cmd\eqL@frame@prevcmd
4098   \global\advance\eqL@column@\@ne
4099   \eqL@columns@print@cell
4100 }

```

lums@print@line@end

```

4101 \def\eqL@columns@print@line@end{%
4102   \eqL@punct@apply@line
4103   \eqL@hook@lineout
4104 % \TODO add an even column with empty stuff if box processing deferred
4105   \ifodd\eqL@column@
4106     \expandafter\eqL@columns@print@trailright
4107   \fi
4108   \eqL@columns@completerow
4109   \eqL@columns@print@tag
4110 }

```

qL@columns@print@tag

```

4111 \def\eqL@columns@print@tag{%
4112   \kern-\dimexpr\eqL@totalwidth+\eqL@colsep@\relax

```

determine first line available space

```

4113 \eqL@display@firstavail@set\eqL@line@avail@
4114 \eqL@columns@overfull
4115 \eqL@numbering@print@line@eval
4116 \if@eqnsw
4117   \eqL@tagbox@make\eqL@composetag@print
4118 \fi
4119 \eqL@tagpos@print@line@eval
4120 \eqL@tagbox@print@cell
4121 \eqL@tagpos@print@line@end
4122 }

```

\eqL@columns@print

```

4123 \def\eqL@columns@print{%
4124 (dev)\eqL@dev@enter\eqL@columns@print
4125   \eqL@shape@align@disable
4126   \eqL@display@halign@init\eqL@columns@print@line@begin
4127   \eqL@display@halign@letcr\eqL@columns@print@line@end
4128   \tabskip\eqL@marginleft@

4129 \halign{%
4130   &%
4131   \global\advance\eqL@column@\@ne
4132   \global\let\eqL@cell@container\@empty
4133   \global\setbox\eqL@cellbox@\hbox{%

```

```

4134 \eql@strut@cell
4135 \@lign
4136 $\m@th\displaystyle
4137 \eql@hook@colin
4138 ##%
4139 \eql@class@innerleft
4140 \eql@hook@innerleft
4141 \eql@tagging@mathsave
4142 $%
4143 \eql@tagging@mathaddlast
4144 }%
4145 \eql@cell@container
4146 \hfil
4147 \eql@columns@print@cell
4148 \global\let\eql@frame@prevcmd\eql@frame@cmd
4149 \tabskip\z@skip
4150 &%
4151 \eql@prevwidth@wd\eql@cellbox@
4152 \let\eql@frame@cmd\eql@frame@prevcmd
4153 \global\advance\eql@column@\@ne
4154 \global\let\eql@cell@container\@empty
4155 \setbox\eql@cellbox@\hbox{%
4156 \unhbox\eql@cellbox@
4157 \eql@strut@cell
4158 \@lign
4159 $\m@th\displaystyle
4160 \eql@hook@innerright
4161 \eql@class@innerright@sel
4162 ##%
4163 \eql@punct@apply@col
4164 \eql@hook@colout
4165 \eql@tagging@mathsave
4166 $%
4167 \eql@tagging@mathaddlast
4168 }%
4169 \eql@cell@container
4170 \eql@columns@print@cell
4171 \hfil
4172 \tabskip\eql@colsep@\relax
4173 \crr

4174 \noalign{%
4175 \eql@display@halign@start
4176 \eql@numbering@print@block@begin
4177 \eql@hook@blockbefore
4178 }%
4179 \eql@hook@blockin
4180 \eql@scan@body
4181 \ifvmode\else
4182 \relax
4183 \eql@punct@apply@block
4184 \eql@hook@blockout
4185 \eql@display@endline
4186 \cr
4187 \fi
4188 \noalign{%
4189 \eql@hook@blockafter
4190 \eql@display@halign@end
4191 (dev)\eql@dev@leave\eql@columns@print

```



```

4192   }%
4193   \eql@tagging@tablesavealign
4194 }%
4195 }

```

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.

```
4196 \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:

```

4197 \def\eql@scan@body@dump{\the\eql@scan@reg@}
4198 \def\eql@scan@body@rescan{%
4199   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
4200 \let\eql@scan@body\eql@scan@body@dump

```

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

```
4201 \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@`:

```

4202 \def\eql@scan@env#1{%
4203   (dev)\eql@dev@enter\eql@scan@env
4204   \def\eql@scan@end{#1}\expandafter\end\expandafter{\@currenvir}}%
4205   \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`:

```

4206   \edef\eql@scan@iterate{\expandafter\noexpand\curname\@currenvir\endcurname}%
4207   \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate

```

```

4208 \ifdefined\eql@scan@par
4209   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
4210 \else
4211   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
4212 \fi
4213 \eql@scan@iterate
4214 }

```

`\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:

```

4215 \long\def\eql@scan@env@iterate#1\end#2{%
4216   \edef\eql@scan@stack{%
4217     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4218   \ifx\@empty\eql@scan@stack
4219     \@checkend{#2}%
4220     \eql@scan@addto{#1}%
4221     \expandafter\let\eql@scan@iterate\eql@scan@env@org
4222 (dev)\eql@dev@leave\eql@scan@env
4223   \expandafter\eql@scan@end
4224 \else
4225   \eql@scan@addto{#1\end{#2}}%
4226   \expandafter\eql@scan@iterate
4227 \fi
4228 }

```

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

```

4229 \def\eql@scan@env@iterate@nopar#1\end#2{\eql@scan@env@iterate#1\end{#2}}

```

`\eql@scan@env@count` When adding a piece of the current environment's contents to `\eql@scan@reg@`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

4230 \long\def\eql@scan@env@count#1\begin#2{%
4231   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4232 }

```

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

```

4233 \def\eql@scan@env@cancel{%
4234   \@namedef{end\@currentvir}{\ignorespacesafterend}%
4235 }

```

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 `\]`:

```

4236 \def\eql@scan@sqr#1{%
4237 (dev)\eql@dev@enter\eql@scan@sqr
4238   \def\eql@scan@end{#1\]}%
4239   \eql@scan@reg@{\def\eql@scan@stack{b}}%

```

```

4240 \let\eql@scan@sqr@org\[%\]
4241 \ifdefined\eql@scan@par
4242   \let\[\eql@scan@sqr@iterate%\]
4243 \else
4244   \let\[\eql@scan@sqr@iterate@nopar%\]
4245 \fi
4246 \[%\]
4247 }

```

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

```

4248 \long\def\eql@scan@sqr@iterate#1\{%
4249   \edef\eql@scan@stack{%
4250     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4251   \ifx\@empty\eql@scan@stack
4252     \let\[\eql@scan@sqr@org%\]
4253     \eql@scan@addto{#1}%
4254 (dev)\eql@dev@leave\eql@scan@sqr
4255   \expandafter\eql@scan@end
4256 \else
4257   \eql@scan@addto{#1\]}%
4258   \expandafter\[%\]
4259 \fi
4260 }

```

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

```

4261 \def\eql@scan@sqr@iterate@nopar#1\{\eql@scan@sqr@iterate#1\}

```

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

```

4262 \long\def\eql@scan@sqr@count#1\[#2{\]
4263   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4264 }

```

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

```

4265 \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 `\>`:

```

4266 \def\eql@scan@ang#1{%
4267 (dev)\eql@dev@enter\eql@scan@ang
4268   \def\eql@scan@end{#1\>}%
4269   \eql@scan@reg@{\}\def\eql@scan@stack{b}%
4270   \let\eql@scan@ang@org\<%\>
4271   \ifdefined\eql@scan@par
4272     \let\<\eql@scan@ang@iterate%\>
4273 \else
4274     \let\<\eql@scan@ang@iterate@nopar%\>
4275 \fi
4276 \<%\>
4277 }

```

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

```

4278 \long\def\eql@scan@ang@iterate#1\>{%
4279   \edef\eql@scan@stack{%
4280     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4281   \ifx\@empty\eql@scan@stack
4282     \let\<\eql@scan@ang@org%\>
4283     \eql@scan@addto{#1}%
4284 (dev)\eql@dev@leave\eql@scan@ang
4285   \expandafter\eql@scan@end
4286   \else
4287     \eql@scan@addto{#1\>}%
4288     \expandafter\<%\>
4289   \fi
4290 }

```

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

```

4291 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}

```

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

```

4292 \long\def\eql@scan@ang@count#1\<#2\>%
4293   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4294 }

```

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`. All arguments are scanned such that any spaces stop the scanning and such that any alignment markers ‘&’ cannot interfere: **TODO**: update

```

4295 \def\eql@equations@testall{\eql@parseopt\eql@equations@parseopt}
4296 \def\eql@equations@parseopt{%
4297   \ifx\eql@parseopt@token*%
4298     \let\eql@parseopt@next\eql@parseopt@nonumber
4299   \fi
4300   \ifx\eql@parseopt@token!%
4301     \let\eql@parseopt@next\eql@parseopt@donumber
4302   \fi
4303   \ifx\eql@parseopt@token/%
4304     \let\eql@parseopt@next\eql@parseopt@transpose
4305   \fi
4306   \ifx\eql@parseopt@token[%]
4307     \let\eql@parseopt@next\eql@parseopt@opt
4308   \fi
4309   \ifx\eql@parseopt@token\eql@atxi
4310     \let\eql@parseopt@next\eql@parseopt@label
4311   \fi
4312   \ifx\eql@parseopt@token\eql@atxii
4313     \let\eql@parseopt@next\eql@parseopt@label
4314   \fi
4315   \ifx\eql@parseopt@token.%
4316     \let\eql@parseopt@next\eql@parseopt@punctdot
4317   \fi
4318   \ifx\eql@parseopt@token,%
4319     \let\eql@parseopt@next\eql@parseopt@punctcomma
4320   \fi
4321   \ifx\eql@parseopt@token~%

```

```

4322   \let\eql@parseopt@next\eql@parseopt@punctoff
4323   \fi
4324   \ifx\eql@parseopt@token-
4325     \let\eql@parseopt@next\eql@parseopt@single
4326   \fi
4327   \ifx\eql@parseopt@token=
4328     \let\eql@parseopt@next\eql@parseopt@lines
4329   \fi
4330   \ifx\eql@parseopt@token|
4331     \let\eql@parseopt@next\eql@parseopt@columns
4332   \fi
4333   \ifx\eql@parseopt@token\label
4334     \let\eql@parseopt@next\eql@parseopt@end
4335   \fi
4336   \ifx\eql@parseopt@token\begin
4337     \let\eql@parseopt@next\eql@parseopt@end
4338   \fi
4339 }

```

`\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:

```

4340 \def\eql@equations@processopt{%
4341   \let\eql@tags@container@block\eql@tags@container@clear
4342   \let\eql@tags@frame@cmd\@firstofone
4343   \let\eql@skip@force@above\@undefined
4344   \let\eql@skip@force@below\@undefined
4345   \let\eql@skip@force@leave\@undefined
4346   \let\eql@display@linewidth\@undefined
4347   \let\eql@display@marginleft\@undefined
4348   \let\eql@display@marginright\@undefined
4349   \eql@abovespace@\z@skip
4350   \eql@belowspace@\z@skip
4351   \eql@displaybreak@prepen@\@MM
4352   \eql@displaybreak@postpen@\@MM
4353   \eql@nextopt@process{equations}%
4354   \let\eql@punct@block\eql@punct@main
4355   \let\eql@punct@main\relax
4356   \eql@indent@\glueexpr\eql@indent@val\relax
4357   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4358 }

```

O.3 Single-Line Main

In the following, we define the main routine for the single-line equation mode.

`\eql@single@cr` Cannot use line breaks, produce an error message:

```

4359 \def\eql@single@cr{%
4360   \eql@error{Cannot use '\string\` within display equation.
4361     Please switch to equations environment}%
4362 }

```

`\eql@single@start` Opening code for single-line equation. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of

global registers, set native vs. manual equation tag mode, install error message for using `\.`. Hand over to mode-specific opening:

```

4363 \def\eq@single@start{%
4364   \eq@display@center
4365   \eq@tagging@start
4366   \eq@dollar@dollar@begin
4367   \eq@display@adjust
4368   \eq@numbering@init
4369   \eq@stack@save@equations
4370   \eq@numbering@single@init
4371   \ifdefined\eq@single@crerror\else
4372     \let\.\eq@single@cr
4373   \fi
4374   \ifdefined\eq@single@native
4375     \let\eq@single@start@sel\eq@single@start@native
4376     \let\eq@single@end@sel\eq@single@end@native
4377   \else
4378     \let\eq@single@start@sel\eq@single@start@print
4379     \let\eq@single@end@sel\eq@single@end@print
4380   \fi
4381   \eq@single@start@sel
4382 }

```

`\eq@single@end` Closing code for single-line equation. Apply punctuation for the block, perform mode-specific ending, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```

4383 \def\eq@single@end{%
4384   \eq@punct@apply@block
4385   \eq@hook@eqout
4386   \eq@single@end@sel
4387   \eq@stack@restore
4388   \eq@dollar@dollar@end
4389   \eq@tagging@end
4390   \eq@display@leave
4391 }

```

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

```

4392 \def\eq@single@main{%
4393   \expandafter\eq@single@start
4394   \eq@scan@body
4395   \eq@single@end
4396 }

```

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

```

4397 \def\eq@mode@single{%
4398   \ifdefined\eq@single@doscans
4399     \let\eq@equations@main\eq@single@main
4400     \let\eq@equations@end\@empty
4401   \else
4402     \let\eq@equations@main\@undefined
4403     \let\eq@equations@end\eq@single@end
4404   \fi
4405 }

```

O.4 Multi-Line Main

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

```
4406 \let\eql@multi@mode@lines\eql@false
```

`\eql@multi@main` Main routine for multi-line modes. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, initialise macros for use within equations: **TODO:** shove depends on lines vs columns

```
4407 \def\eql@multi@main{%
4408   \eql@display@enter
4409   \eql@tagging@start
4410   \eql@dollar@dollar@begin
4411   \eql@display@adjust
4412   \eql@numbering@init
4413   \eql@stack@save@equations
4414   \ifdefined\eql@transpose@active
4415     \ifdefined\eql@multi@mode@lines\else
4416       \eql@transpose
4417     \fi
4418   \fi
4419   \ifdefined\eql@numbering@subeq@use
4420     \eql@numbering@subeq@init
4421   \fi
4422   \eql@display@init
4423   \let\intertext\eql@intertext
4424   \let\endintertext\endeql@intertext
4425   \eql@shape@align@enable
```

Now measure the given multi-line equations body:

```
4426   \ifdefined\eql@multi@mode@lines
4427     \eql@lines@measure
4428   \else
4429     \ifdefined\eql@ampproof@active
4430       \eql@ampproof
4431     \fi
4432     \eql@columns@measure
4433   \fi
```

If only a single equation number is used for subequation numbering, revert to normal equation numbering. If only a single column is used in columns mode, may fallback to lines mode. Switching from columns to lines mode, the width can be incorrect, expect only minor discrepancies, but for accurateness, should call `\eql@lines@measure`:

```
4434   \ifdefined\eql@numbering@subeq@use
4435     \eql@numbering@subeq@test
4436   \fi
4437   \ifdefined\eql@multi@mode@lines\else
4438     \ifdefined\eql@multi@lines@fallback
4439       \ifnum\eql@totalcolumns@=\@ne
4440         \let\eql@multi@mode@lines\eql@true
4441       \ifx\eql@multi@lines@fallback\z@\else
4442         \eql@lines@measure
4443       \fi
4444     \fi
4445   \fi
4446 \fi
```

Adjust the multi-line equations body:

```
4447 \ifdefined\eql@multi@mode@lines
4448   \eql@lines@adjust
4449 \else
4450   \eql@columns@adjust
4451 \fi
```

Now print the multi-line equations body:

```
4452 \eql@display@print
4453 \eql@numbering@print@init
4454 \ifdefined\eql@multi@mode@lines
4455   \eql@lines@print
4456 \else
4457   \eql@columns@print
4458 \fi
4459 \eql@display@close
```

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

```
4460 \ifdefined\eql@numbering@subeq@use
4461   \eql@numbering@subeq@close
4462 \fi
4463 \eql@stack@restore
4464 \eql@dollar@end
4465 \eql@tagging@end
4466 \eql@display@leave
4467 }
```

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

```
\eql@mode@lines
4468 \def\eql@mode@columns{%
4469   \let\eql@equations@main\eql@multi@main
4470   \let\eql@equations@end\@empty
4471   \let\eql@multi@mode@lines\eql@false
4472 }
4473 \def\eql@mode@lines{%
4474   \let\eql@equations@main\eql@multi@main
4475   \let\eql@equations@end\@empty
4476   \let\eql@multi@mode@lines\eql@true
4477 }
```

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`:

```
4478 \newenvironment{equations}{%
4479 <dev>\eql@dev@enterenv
4480   \ifmmode
4481     \eql@error@mathmode{\string\begin{\@currenvir}}%
```



```

4482   \expandafter\eq@scan@env\expandafter\eq@scan@env@cancel
4483   \else
4484     \expandafter\eq@ampprotect\expandafter\eq@equations@testall
4485     \expandafter\eq@equations@start
4486   \fi
4487 }{%
4488   \eq@equations@end
4489   \ignorespacesafterend
4490 (dev)\eq@dev@leaveenv
4491 }
4492 \eq@markline@amsthm@register{equations}

```

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

```

4493 \def\eq@equations@start{%
4494   \eq@equations@processopt
4495   \ifdefined\eq@equations@main
4496     \expandafter\eq@scan@env\expandafter\eq@equations@main
4497   \else
4498     \expandafter\eq@single@start
4499   \fi
4500 }

```

Square Brackets.

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

```

4501 \newenvironment{equations@sqr}{}{}
4502 \eq@markline@amsthm@register{equations@sqr}

```

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

```

4503 \protected\def\eq@equations@sqr@open{%
4504   \ifmmode
4505     \eq@error@mathmode{\string\[...\string\]}%
4506     \expandafter\eq@scan@sqr\expandafter\eq@scan@sqrang@cancel
4507   \else
4508 (dev)\eq@dev@enter{\[...\string\]}%
4509     \expandafter\eqnaddopt\expandafter{\eq@equations@sqr@opt}%
4510     \begin{equations@sqr}%
4511     \let\]\eq@equations@sqr@close
4512     \expandafter\eq@ampprotect\expandafter\eq@equations@testall
4513     \expandafter\eq@equations@sqr@start
4514   \fi
4515 }

```

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

```

4516 \def\eq@equations@sqr@start{%
4517   \eq@equations@processopt
4518   \ifdefined\eq@equations@main
4519     \expandafter\eq@scan@sqr\expandafter\eq@equations@main

```

```

4520 \else
4521   \expandafter\eql@single@start
4522 \fi
4523 }

```

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

```

4524 \protected\def\eql@equations@sqr@close{%
4525   \eql@equations@end
4526 (dev)\eql@dev@leave{\[...\string\]}%
4527 \end{equations@sqr}%
4528 \ignorespaces
4529 }

```

TODO: describe

```

\eql@sqr@open
\eql@sqr@close
4530 \let\eql@sqr@open\eql@equations@sqr@open
4531 \protected\def\eql@sqr@close{%
4532   \eql@error{'\string\'} may only close '\string\[\'%
4533 }

```

Angle Brackets.

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

```

4534 \newenvironment{equations@ang}{}{}
4535 \newenvironment{equationsbox@ang}{}{}
4536 \eql@markline@amsthm@register{equations@ang}

```

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

```

4537 \protected\def\eql@ang@open{%
4538 (dev)\eql@dev@enter{\<...\string\>}%
4539   \ifmmode
4540     \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4541     \begin{equationsbox@ang}%
4542     \let\>\eql@box@ang@close
4543     \expandafter\eql@ampprotect\expandafter\eql@box@testall
4544     \expandafter\eql@box@start
4545   \else
4546     \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
4547     \begin{equations@ang}%
4548     \let\>\eql@equations@ang@close
4549     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4550     \expandafter\eql@equations@ang@start
4551   \fi
4552 }

```

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

```

4553 \protected\def\eql@ang@close{%
4554   \eql@error{'\string\>} may only close '\string\<\'%
4555 }

```

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

```

4556 \def\eqlequations@ang@start{%
4557   \eqlequations@processopt
4558   \ifdefined\eqlequations@main
4559     \expandafter\eql@scan@ang\expandafter\eqlequations@main
4560   \else
4561     \expandafter\eql@single@start
4562   \fi
4563 }

```

`\eqlequations@ang@close` **TODO:** describe

```

4564 \def\eqlequations@ang@close{%
4565   \eqlequations@end
4566   \end{equations@ang}%
4567 (dev)\eql@dev@leave{\<...\string\>}%
4568   \ignorespaces
4569 }

```

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

```

4570 \def\eql@box@ang@close{%
4571   \eql@box@end
4572   \end{equationsbox@ang}%
4573 (dev)\eql@dev@leave{\<...\string\>}%
4574   \ignorespaces
4575 }

```

P Options

P.1 Selection Tools

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

```

4576 \def\eql@decide@abovebelow#1#2#3#4#5{%
4577   \eql@decide@select{#1}{#2}{#3}{%
4578     {,abovebelow,both,tb}{#4#5},%
4579     {above,top,t}{#4},%
4580     {below,bottom,b}{#5}}

```

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

```

4581 \def\eql@decide@situation#1#2#3#4{%
4582   \eql@decide@select{#1}{#2}{#3}{%
4583     {{long}{\def#4{0}}},%
4584     {{short}{\def#4{1}}},%
4585     {{cont}{\def#4{2}}},%
4586     {{par}{\def#4{3}}},%
4587     {{top}{\def#4{4}}},%
4588     {{noskip}{\def#4{5}}},%
4589     {{medskip}{\def#4{6}}}}

```

P.2 Options Declarations

We now declare all key-value pairs for options sorted by their category.

Modes for Equations Box Environment. Declare horizontal and vertical alignment modes for the boxed equations environment. Also declare spacing of columns:

```

4590 \eql@define@key{equationsbox}{gathered,gather,ga,lines,ln}[][%
4591   \eql@mode@stacked}
4592 \eql@define@key{equationsbox}{aligned,align,al,columns,col}[][%
4593   \eql@mode@aligned}
4594 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
4595 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
4596 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
4597 \eql@define@key{setup}{boxangopt}[]{%
4598   \def\eql@box@ang@opt{columns,#1}}

```

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

```

4599 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@single}
4600 \eql@define@key{equations}{gathered,gather,ga,lines,ln}[][%
4601   \eql@mode@lines}
4602 \eql@define@key{equations}{aligned,align,al,columns,col}[][%
4603   \eql@mode@columns}
4604 \eql@define@key{equations,setup}{transpose}[true]{%
4605   \eql@decide@select{#3}{#2}{#1}{%
4606     {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
4607     {\noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
4608     {\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}}
4609 \eql@define@key{equations}{native}[true]{%
4610   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
4611   \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
4612 \eql@define@key{setup}{native}[true]{%
4613   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
4614 \eql@define@key{setup}{scanequation}[true]{%
4615   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscans}
4616 \eql@define@key{setup}{sqropt}[]{%
4617   \def\eql@equations@sqropt{equation,#1}}
4618 \eql@define@key{setup}{angopt}[]{%
4619   \def\eql@equations@ang@opt{columns,#1}}

```

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

```

4620 \def\eql@keycat{equations,equationsbox,setup}
4621 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
4622 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
4623   {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
4624   {{cell}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
4625   {{tag}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
4626   {\eql@decide@true
4627     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
4628 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings concerning page breaks:

```

4629 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4630   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
4631   {{high,3}{\eql@displaybreak@pre3}},%
4632   {{med,medium,2}{\eql@displaybreak@pre2}},%
4633   {{low,1}{\eql@displaybreak@pre1}},%

```

```

4634     {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
4635     {{default,inherit,-1}{\eql@displaybreak@pre\m@ne}}}}
4636 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}}{%
4637     {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
4638     {{high,3}{\eql@displaybreak@post3}},%
4639     {{med,medium,2}{\eql@displaybreak@post2}},%
4640     {{low,1}{\eql@displaybreak@post1}},%
4641     {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
4642     {{default,inherit,-1}{\eql@displaybreak@post\m@ne}}}}
4643 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4644     \eql@decide@select{#3}{#2}{#1}}{%
4645     {{full,4}{\eql@displaybreak@inter4}},%
4646     {{high,3}{\eql@displaybreak@inter3}},%
4647     {{med,medium,2}{\eql@displaybreak@inter2}},%
4648     {{low,1}{\eql@displaybreak@inter1}},%
4649     {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
4650 \eql@define@key{equations}{prepenalty}{%
4651     \eql@displaybreak@prepen@numexpr#1\relax}
4652 \eql@define@key{equations}{postpenalty}{%
4653     \eql@displaybreak@postpen@numexpr#1\relax}
4654 \eql@define@key{equations,setup}{interpenalty}{%
4655     \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

4656 \eql@define@key{control}{vspace}[]{\eql@vspace@add{#1}}
4657 \eql@define@key{control}{vspace*}[]{\eql@vspace@addfixedbefore{#1}}
4658 \eql@define@key{control}{vspace!}[]{\eql@vspace@addfixedafter{#1}}
4659 \eql@define@key{control}{break}[4]{\eql@displaybreak@level[#{#1}]}
4660 \eql@define@key{control}{penalty}[]{\eql@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4661 \eql@define@key\eql@keycat{displayheight}[strut]{%
4662     \eql@decide@select{#3}{#2}{#1}}{%
4663     {\eql@decide@false{\let\eql@display@height\@undefined}},%
4664     {{strut}{\def\eql@display@height{\ht\eql@strutbox@}}},%
4665     {\relax{\def\eql@display@height{#1}}}}
4666 \eql@define@key\eql@keycat{displaydepth}[strut]{%
4667     \eql@decide@select{#3}{#2}{#1}}{%
4668     {\eql@decide@false{\let\eql@display@depth\@undefined}},%
4669     {{strut}{\def\eql@display@depth{\dp\eql@strutbox@}}},%
4670     {\relax{\def\eql@display@depth{#1}}}}

```

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

```

4671 \eql@define@key{equations}{noskip}[]{%
4672     \eql@decide@abovebelow{#3}{#2}{#1}}%
4673     {\def\eql@skip@force@above{5}}%
4674     {\def\eql@skip@force@below{5}}}}
4675 \eql@define@key{equations}{short}[above]{%
4676     \eql@decide@abovebelow{#3}{#2}{#1}}%
4677     {\def\eql@skip@force@above{1}}%
4678     {\def\eql@skip@force@below{1}}}}
4679 \eql@define@key{equations}{long}[]{%
4680     \eql@decide@abovebelow{#3}{#2}{#1}}%
4681     {\def\eql@skip@force@above{0}}%
4682     {\def\eql@skip@force@below{0}}}}
4683 \eql@define@key{equations}{medskip}[]{%

```

```

4684 \eql@decide@abovebelow{#3}{#2}{#1}%
4685   {\def\eql@skip@force@above{6}}%
4686   {\def\eql@skip@force@below{6}}%
4687 \eql@define@key{equations}{par}[par]{%
4688   \eql@decide@select{#3}{#2}{#1}{%
4689     {{default,}\let\eql@skip@force@leave\undefined}},%
4690     {{cont,hmode}{\let\eql@skip@force@leave\z@}},%
4691     {{par,vmode}{\let\eql@skip@force@leave\@ne
4692       \ifdefined\eql@skip@force@below\else
4693         \def\eql@skip@force@below{3}%
4694       \fi}},%
4695     {{top}{\let\eql@skip@force@leave\tw@
4696       \ifdefined\eql@skip@force@below\else
4697         \def\eql@skip@force@below{4}
4698       \fi}}}}

```

Specify vertical spacing explicitly:

```

4699 \eql@define@key{equations}{skip}{%
4700   \def\eql@skip@force@above{7}%
4701   \def\eql@skip@custom@above{#1}%
4702   \let\eql@skip@force@below\eql@skip@force@above
4703   \let\eql@skip@custom@below\eql@skip@custom@above}
4704 \eql@define@key{equations}{aboveskip}{%
4705   \def\eql@skip@force@above{7}%
4706   \def\eql@skip@custom@above{#1}}
4707 \eql@define@key{equations}{belowskip}{%
4708   \def\eql@skip@force@below{7}%
4709   \def\eql@skip@custom@below{#1}}
4710 \eql@define@key{equations}{abovespace}{%
4711   \advance\eql@abovespace@\glueexpr#1\relax}
4712 \eql@define@key{equations}{belowspace}{%
4713   \advance\eql@belowspace@\glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4714 \eql@define@key{intertext}{skip}{%
4715   \def\eql@skip@force@above{7}%
4716   \def\eql@skip@custom@above{#1}%
4717   \let\eql@skip@force@below\eql@skip@force@above
4718   \let\eql@skip@custom@below\eql@skip@custom@above}
4719 \eql@define@key{intertext}{aboveskip}{%
4720   \def\eql@skip@force@below{7}%
4721   \def\eql@skip@custom@below{#1}}
4722 \eql@define@key{intertext}{belowskip}{%
4723   \def\eql@skip@force@above{7}%
4724   \def\eql@skip@custom@above{#1}}
4725 \eql@define@key{intertext}{noskip}[]{%
4726   \eql@decide@abovebelow{#3}{#2}{#1}%
4727   {\def\eql@skip@force@below{5}}%
4728   {\def\eql@skip@force@above{5}}}
4729 \eql@define@key{intertext}{short}[]{%
4730   \eql@decide@abovebelow{#3}{#2}{#1}%
4731   {\def\eql@skip@force@below{1}}%
4732   {\def\eql@skip@force@above{1}}}
4733 \eql@define@key{intertext}{long}[]{%
4734   \eql@decide@abovebelow{#3}{#2}{#1}%
4735   {\def\eql@skip@force@below{0}}%
4736   {\def\eql@skip@force@above{0}}}
4737 \eql@define@key{intertext}{medskip}[]{%

```

```

4738 \eql@decide@abovebelow{#3}{#2}{#1}%
4739   {\def\eql@skip@force@below{6}}%
4740   {\def\eql@skip@force@above{6}}

```

Configure general vertical spacing behaviour for various situations:

```

4741 \eql@define@key{setup}{skip, longskip}{%
4742   \abovedisplayskip\glueexpr#1\relax
4743   \belowdisplayskip\abovedisplayskip
4744   \def\eql@skip@long@above{#1}%
4745   \let\eql@skip@long@below\eql@skip@long@above}
4746 \eql@define@key{setup}{aboveskip, abovelongskip}{%
4747   \abovedisplayskip\glueexpr#1\relax
4748   \def\eql@skip@long@above{#1}}
4749 \eql@define@key{setup}{belowskip, belowlongskip}{%
4750   \belowdisplayskip\glueexpr#1\relax
4751   \def\eql@skip@long@below{#1}}
4752 \eql@define@key{setup}{aboveshortskip}{%
4753   \abovedisplayshortskip\glueexpr#1\relax
4754   \def\eql@skip@short@above{#1}}
4755 \eql@define@key{setup}{belowshortskip}{%
4756   \belowdisplayshortskip\glueexpr#1\relax
4757   \def\eql@skip@short@below{#1}}
4758 \eql@define@key{setup}{tagskip}{%
4759   \def\eql@skip@tag@above{#1}%
4760   \let\eql@skip@tag@below\eql@skip@tag@above}
4761 \eql@define@key{setup}{abovetagskip}{%
4762   \def\eql@skip@tag@above{#1}}
4763 \eql@define@key{setup}{belowtagskip}{%
4764   \def\eql@skip@tag@below{#1}}
4765 \eql@define@key{setup}{medskip}{%
4766   \def\eql@skip@med@above{#1}%
4767   \let\eql@skip@med@below\eql@skip@med@above}
4768 \eql@define@key{setup}{abovemedskip}{%
4769   \def\eql@skip@med@above{#1}}
4770 \eql@define@key{setup}{belowmedskip}{%
4771   \def\eql@skip@med@below{#1}}
4772 \eql@define@key{setup}{abovetopskip}{%
4773   \def\eql@skip@top@above{#1}}
4774 \eql@define@key{setup}{belowtopskip}{%
4775   \def\eql@skip@top@below{#1}}
4776 \eql@define@key{setup}{aboveparskip}{%
4777   \def\eql@skip@par@above{#1}}
4778 \eql@define@key{setup}{belowparskip}{%
4779   \def\eql@skip@par@below{#1}}
4780 \eql@define@key{setup}{abovecontskip}{%
4781   \eql@decide@select{#3}{#2}{#1}{%
4782     {{hide}}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%
4783     {\relax{\def\eql@skip@cont@above{#1}}}}}}
4784 \eql@define@key{setup}{belowcontskip}{%
4785   \def\eql@skip@cont@below{#1}}
4786 \eql@define@key{setup}{shortmode}{%
4787   \eql@decide@select{#3}{#2}{#1}{%
4788     {{off, never, no}}{\def\eql@skip@mode@short{0}}},%
4789     {{above, neverbelow, notbelow, belowoff}}{\def\eql@skip@mode@short{1}}},%
4790     {{belowone, belowsingle}}{\def\eql@skip@mode@short{2}}},%
4791     {{belowall, always, on}}{\def\eql@skip@mode@short{3}}}}}}
4792 \eql@define@key{setup}{abovecontmode}{%
4793   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}

```

```

4794 \eql@define@key{setup}{belowcontmode}{%
4795   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
4796 \eql@define@key{setup}{aboveparmode}{%
4797   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
4798 \eql@define@key{setup}{belowparmode}{%
4799   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
4800 \eql@define@key{setup}{abovetopmode}{%
4801   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
4802 \eql@define@key{setup}{belowtopmode}{%
4803   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

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

```

4804 \def\eql@keycat{equations,subequations}
4805 \eql@define@key\eql@keycat{label}{\eql@tags@addblock@label{#1}}
4806 \eql@define@key\eql@keycat{labelname}{\eql@tags@addblock@name{#1}}
4807 \eql@define@key\eql@keycat{tag}{\eql@tags@addblock@tag{#1}}
4808 \eql@define@key\eql@keycat{tag*}{%
4809   \eql@tags@addblock@tagform@off\eql@tags@addblock@tag{#1}}
4810 \eql@define@key\eql@keycat{taglabel}{\eql@tags@addblock@ref{#1}}

```

TODO: describe

```

4811 \eql@define@key{control}{label}{\eql@tags@add@label{#1}}
4812 \eql@define@key{control}{labelname}{\eql@tags@add@name{#1}}
4813 \eql@define@key{control}{tag}{\eql@tags@add@tag{#1}}
4814 \eql@define@key{control}{tag*}{\eql@tags@add@tagform@off\eql@tags@add@tag{#1}}
4815 \eql@define@key{control}{taglabel}{\eql@tags@add@ref{#1}}
4816 \eql@define@key{control}{shifftag}{\eql@tags@add@raiseshift{#1}}
4817 \eql@define@key{control}{smashtag}{\eql@tags@add@raisesmash{#1}}
4818 \eql@define@key{control}{pushtag}[]{\eql@tags@add@forceraise}

```

TODO: describe

```

4819 \eql@define@key{setup}{labelname}{\protected@edef\eql@tags@name@generic{#1}}
4820 \eql@define@key{setup}{autolabel}[true]{%
4821   \eql@decide@bool{#3}{#2}{#1}\eql@tags@autolabel}
4822 \eql@define@key{setup}{autotag}[true]{%
4823   \eql@decide@bool{#3}{#2}{#1}\eql@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

4824 \def\eql@keycat{equations,setup}
4825 \eql@define@key\eql@keycat{tagmargin}[auto]{%
4826   \eql@decide@select{#3}{#2}{#1}{%
4827     {{auto,\eql@decide@false}{\let\eql@tagmargin@val\undefined}},%
4828     {\relax{\def\eql@tagmargin@val{#1}}}}}
4829 \eql@define@key\eql@keycat{tagmargin*}{%
4830   \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}}
4831 \eql@define@key\eql@keycat{tagmarginratio}{%
4832   \eql@tagmargin@ratio@{dimexpr#1pt\relax}}
4833 \eql@define@key\eql@keycat{tagmarginthreshold}{%
4834   \def\eql@tagmargin@threshold{#1}}
4835 \eql@define@key\eql@keycat{mintagsep}{\def\eql@tagsepmin@val{#1}}
4836 \eql@define@key\eql@keycat{mintagwidth}{%
4837   \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}}
4838 \eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin@{#1}}
4839 \eql@define@key\eql@keycat{tagsnap}{%
4840   \eql@decide@select{#3}{#2}{#1}{%

```



```

4841   {\eql@decide@false{\let\eql@tagpos@snap@z@},%
4842   {\relax{\def\eql@tagpos@snap{#1}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

4843 \def\eql@keycat{equations,setup}
4844 \eql@define@key\eql@keycat{tagbox,taglayout}{%
4845   \eql@tags@taglayout@set{#1}}
4846 \eql@define@key\eql@keycat{tagbox*,taglayout*}{%
4847   \eql@tags@taglayout@set@direct{#1}}
4848 \eql@define@key\eql@keycat{tagform}{%
4849   \eql@tags@tagform@set{#1}}
4850 \eql@define@key\eql@keycat{tagform*}{%
4851   \eql@tags@tagform@set@direct{#1}}
4852 \eql@define@key\eql@keycat{subeqtemplate}{%
4853   \def\eql@subequations@template###1###2{#1}%
4854   \eql@append\eql@subequations@template{\theparentequation{equation}}}}

4855 \eql@define@key{control}{tagbox,taglayout}{%
4856   \global\eql@append\eql@tags@container{\eql@tags@taglayout@set{#1}}}
4857 \eql@define@key{control}{tagbox*,taglayout*}{%
4858   \global\eql@append\eql@tags@container{\eql@tags@taglayout@set@direct{#1}}}
4859 \eql@define@key{control}{tagform}{%
4860   \global\eql@append\eql@tags@container{\eql@tags@tagform@set{#1}}}
4861 \eql@define@key{control}{tagform*}{###1}{%
4862   \global\eql@append\eql@tags@container{\eql@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

4863 \def\eql@keycat{equations,setup}
4864 \eql@define@key\eql@keycat{numberline,number,num,numline,n}[all]{%
4865   \eql@decide@select{#3}{#2}{#1}{%
4866     {\eql@decide@false,0,*}{\let\eql@numbering@active\eql@false}},%
4867     {\eql@decide@true,!}{\let\eql@numbering@active\eql@true}},%
4868     {none,n,-}{\let\eql@numbering@mode\eql@numbering@mode@multi
4869       \let\eql@numbering@active\eql@false}},%
4870     {single,1}{\let\eql@numbering@mode\eql@numbering@mode@single
4871       \let\eql@numbering@active\eql@true}},%
4872     {multi,@}{\let\eql@numbering@mode\eql@numbering@mode@multi
4873       \let\eql@numbering@active\eql@true}},%
4874     {\relax{\eql@numbering@set{#1}}}}
4875 \eql@define@key\eql@keycat{nonumber,nn,*}[]{%
4876   \let\eql@numbering@active\eql@false}
4877 \eql@define@key\eql@keycat{donumber,dn,!}[]{%
4878   \let\eql@numbering@active\eql@true}
4879 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
4880 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
4881 \eql@define@key\eql@keycat{tags,eqno}{%
4882   \eql@decide@select{#3}{#2}{#1}{%
4883     {\right,r}{\let\eql@tagsleft\eql@false}},%
4884     {\left,l}{\let\eql@tagsleft\eql@true}}}}
4885 \eql@define@key\eql@keycat{evadetag,avoidtag}[true]{%
4886   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}
4887 \eql@define@key\eql@keycat{tagbetween}[true]{%
4888   \eql@decide@bool{#3}{#2}{#1}\eql@tagpos@doconvert}

```

TODO: describe

```

4889 \eql@define@key{control}{nonumber,nn,*}[]{\global\@eqnswfalse}

```

```

4890 \eql@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
4891 \eql@define@key{control}{numberhere}[]{\eql@numberhere}
4892 \eql@define@key{control}{numbernext}[]{\eql@numbernext}

```

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

```

4893 \def\eql@keycat{equations,setup}
4894 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
4895   {{center,c}{\let\eql@layoutleft\eql@false}},%
4896   {{left,l}{\let\eql@layoutleft\eql@true}}}}
4897 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
4898 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
4899 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin{#1}}
4900 \eql@define@key\eql@keycat{leftmargin*}{%
4901   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin{\the\dimen@}}
4902 \eql@define@key\eql@keycat{minleftmargin}{%
4903   \def\eql@layoutleftmarginmin{#1}}
4904 \eql@define@key\eql@keycat{maxleftmargin}{%
4905   \eql@decide@select{#3}{#2}{#1}{%
4906     {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
4907     {\relax{\def\eql@layoutleftmarginmax{#1}}}}}

4908 \def\eql@keycat{equations,equationsbox}
4909 \eql@define@key\eql@keycat{margin}{%
4910   \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
4911 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}
4912 \eql@define@key\eql@keycat{marginright}{\def\eql@display@marginright{#1}}
4913 \eql@define@key{equations}{linewidth,width}{\def\eql@display@linewidth{#1}}

```

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

```

4914 \def\eql@keycat{equations,setup}
4915 \eql@define@key\eql@keycat{alignshrink}{\eql@decide@select{#3}{#2}{#1}{%
4916   {{max,full,4}{\eql@alignbadness@inf@bad}},%
4917   {{high,3}{\eql@alignbadness@54\relax}},%
4918   {{med,medium,2}{\eql@alignbadness@18\relax}},%
4919   {{low,1}{\eql@alignbadness@6\relax}},%
4920   {{0,\eql@decide@false}{\eql@alignbadness@z@}}}}
4921 \eql@define@key\eql@keycat{tagshrink}{\eql@decide@select{#3}{#2}{#1}{%
4922   {{max,full,4}{\eql@tagbadness@inf@bad}},%
4923   {{high,3}{\eql@tagbadness@54\relax}},%
4924   {{med,medium,2}{\eql@tagbadness@18\relax}},%
4925   {{low,1}{\eql@tagbadness@6\relax}},%
4926   {{0,\eql@decide@false}{\eql@tagbadness@z@}}}}
4927 \eql@define@key\eql@keycat{alignbadness}{\eql@alignbadness@numexpr#1\relax}
4928 \eql@define@key\eql@keycat{tagbadness}{\eql@tagbadness@numexpr#1\relax}
4929 \eql@define@key\eql@keycat{mincolsep}{\eql@decide@select{#3}{#2}{#1}{%
4930   {{0,\eql@decide@false}{\def\eql@colsepmin{val{0pt}}},%
4931   {\relax{\def\eql@colsepmin{val{#1}}}}}
4932 \eql@define@key\eql@keycat{maxcolsep}{\eql@decide@select{#3}{#2}{#1}{%
4933   {\eql@decide@false{\def\eql@colsepmax{val{.5\maxdimen}}},%
4934   {\relax{\def\eql@colsepmax{val{#1}}}}}
4935 \eql@define@key\eql@keycat{fulllength}[true]{%
4936   \eql@decide@bool{#3}{#2}{#1}\eql@columns@fulllength}

4937 \eql@define@key{equationsbox,setup}{colsep}{\eql@decide@select{#3}{#2}{#1}{%

```

```

4938   {{0,\eql@decide@false}}{\def\eql@box@colsep{Opt}}},%
4939   {\relax{\def\eql@box@colsep{#1}}}}
4940 \eql@define@key{equations}{colsep}{\eql@decide@select{#3}{#2}{#1}}{%
4941   {{0,\eql@decide@false}}{\def\eql@box@colsep{Opt}}},%
4942   {\relax{\def\eql@box@colsep{#1}}}}%
4943   \let\eql@colsepmin@val\eql@box@colsep
4944   \let\eql@colsepmax@val\eql@box@colsep}

```

Horizontal Shape. Configure horizontal alignment schemes:

```

4945 \def\eql@keycat{equations,equationsbox,setup}
4946 \eql@define@key\eql@keycat{shape}[default]{\eql@shape@set{#1}}
4947 \eql@define@key\eql@keycat{padding,pad}[indent]{%
4948   \eql@decide@select{#3}{#2}{#1}}{%
4949   {{max}}{\let\eql@paddingleft@val\undefined}},%
4950   {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
4951   {{0,\eql@decide@false}}{\def\eql@paddingleft@val{Opt}}},%
4952   {\relax{\def\eql@paddingleft@val{#1}}}}%
4953   \let\eql@paddingright@val\eql@paddingleft@val}
4954 \eql@define@key\eql@keycat{padleft}[indent]{%
4955   \eql@decide@select{#3}{#2}{#1}}{%
4956   {{max}}{\let\eql@paddingleft@val\undefined}},%
4957   {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
4958   {{0,\eql@decide@false}}{\def\eql@paddingleft@val{Opt}}},%
4959   {\relax{\def\eql@paddingleft@val{#1}}}}}
4960 \eql@define@key\eql@keycat{padright}[indent]{%
4961   \eql@decide@select{#3}{#2}{#1}}{%
4962   {{max}}{\let\eql@paddingright@val\undefined}},%
4963   {{indent}}{\def\eql@paddingright@val{\eql@indent@val}}},%
4964   {{0,\eql@decide@false}}{\def\eql@paddingright@val{Opt}}},%
4965   {\relax{\def\eql@paddingright@val{#1}}}}}
4966 \eql@define@key\eql@keycat{indent}[2em]{%
4967   \def\eql@indent@val{#1}}

```

TODO: describe

```

4968 \eql@define@key{control}{align}[]{%
4969   \eql@decide@select{#3}{#2}{#1}}{%
4970   {{l,left}}{\global\eql@append\eql@cell@container{\eql@shape@pos@z}}},%
4971   {{c,center}}{\global\eql@append\eql@cell@container{\eql@shape@pos@ne}}},%
4972   {{r,right}}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw}}}}}
4973 \eql@define@key{control}{shift,shifto}[]{%
4974   \eql@decide@select{#3}{#2}{#1}}{%
4975   {{*,indent}}{\eql@shape@alignamount@set{\eql@indent}}},%
4976   {{!,outdent}}{\eql@shape@alignamount@set{-\eql@indent}}},%
4977   {\relax{\eql@shape@alignamount@set{#1}}}}}
4978 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

4979 \def\eql@keycat{equations,equationsbox,setup}
4980 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
4981 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
4982 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
4983 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
4984 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
4985 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}}{%
4986   {{ampeq,amprel,eqafter,beforerel}}\eql@class@ampeq}},%
4987   {{eqamp,relamp,eqbefore,afterrel}}\eql@class@eqamp}}

```

Punctuation. Configure punctuation defaults:

```
4988 \def\eq@keycat{equations,equationsbox,setup}
4989 \eq@define@key\eq@keycat{punctsep}[\,]{\def\eq@punct@sep{#1}}
4990 \eq@define@key\eq@keycat{punct}[.]{\def\eq@punct@main{#1}}
4991 \eq@define@key\eq@keycat{punct*}[]{\let\eq@punct@main\relax}
4992 \eq@define@key\eq@keycat{punctline}[,]{\def\eq@punct@line{#1}}
4993 \eq@define@key\eq@keycat{punctline*}[]{\let\eq@punct@line\relax}
4994 \eq@define@key\eq@keycat{punctcol}[,]{\def\eq@punct@col{#1}}
4995 \eq@define@key\eq@keycat{punctcol*}[]{\let\eq@punct@col\relax}

4996 \eq@define@key{control}{punctsep}[\,]{\def\eq@punct@sep{#1}}
4997 \eq@define@key{control}{punct}[.]{\def\eq@punct@block{#1}%
4998 \def\eq@punct@line{#1}\def\eq@punct@col{#1}}
4999 \eq@define@key{control}{punct*}[]{\let\eq@punct@block\relax}
5000 \eq@define@key{control}{punctapply}[]{\eq@punct@apply@block}
```

Frames. **TODO:** describe

```
5001 \eq@define@key{equationsbox}{frame}[\fbox]{%
5002 \def\eq@box@frame{#1}%
5003 \ifx\eq@box@frame\empty\let\eq@box@frame\firstofone\fi}
5004 \eq@define@key{equationsbox}{wrap}[]{\eq@box@wrap#1}
```

TODO: describe

```
5005 \eq@define@key{control}{framecell}[\fbox]{%
5006 \global\eq@append\eq@cell@container{\def\eq@frame@cmd{#1}}%
5007 \eq@define@key{control}{frametag}[\fbox]{%
5008 \global\eq@append\eq@tags@container{\def\eq@tags@frame@cmd{#1}}}
```

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

```
5009 \eq@define@key{equations,equationsbox}{alt}{}
```

Injections.

```
5010 \eq@define@key{control}{inject}{%
5011 \global\eq@append\eq@interline@container{%
5012 \eq@append\eq@display@injectbefore{#1}}%
5013 \eq@define@key{control}{inject*}{%
5014 \global\eq@append\eq@interline@container{%
5015 \eq@append\eq@display@injectafter{#1}}%
5016 \eq@define@key{control}{markline}[]{\eq@markline@inject{#1}}
5017 \eq@define@key{control}{markline*}[]{\eq@markline@inject{push,#1}}
5018 \eq@define@key{control}{qed}[]{\eq@markline@inject{qed,#1}}
5019 \eq@define@key{control}{qed*}[]{\eq@markline@inject{qed,push,#1}}
```

TODO: describe

```
5020 \eq@define@key{markline}{pos}{%
5021 \eq@decide@select{#3}{#2}{#1}{%
5022 {below,push}{\let\eq@markline@pos\eq@markline@pos@below}},%
5023 {baseline}{\let\eq@markline@pos\eq@markline@pos@baseline}},%
5024 {bottom}{\let\eq@markline@pos\eq@markline@pos@bottom}}%
5025 \eq@define@key{markline}{below,push}[]{%
5026 \let\eq@markline@pos\eq@markline@pos@below}
5027 \eq@define@key{markline}{baseline}[]{%
```

```

5028 \let\eq@markline@pos\eq@markline@pos@baseline}
5029 \eq@define@key{markline}{bottom}[]{%
5030 \let\eq@markline@pos\eq@markline@pos@bottom}
5031 \eq@define@key{markline}{shift}{\def\eq@markline@shift{#1}}
5032 \eq@define@key{markline}{symbol}{\def\eq@markline@symbol{#1}}
5033 \eq@define@key{markline}{qed}[]{\let\eq@markline@symbol\eq@markline@qed}
5034 \eq@define@key{setup}{marksymbol}{\def\eq@markline@symbol{#1}}
5035 \eq@define@key{setup}{qedsymbol}{\def\eq@markline@qed{#1}}
5036 \eq@define@key{setup}{markpos}{%
5037 \eq@decide@select{#3}{#2}{#1}{%
5038   {below}{\let\eq@markline@pos\eq@markline@pos@below}},%
5039   {baseline}{\let\eq@markline@pos\eq@markline@pos@baseline}},%
5040   {bottom}{\let\eq@markline@pos\eq@markline@pos@bottom}}}}

```

Global Switches. Set global switches:

```

5041 \let\eq@multi@linesfallback\eq@false
5042 \let\eq@scan@par\eq@false
5043 \let\eq@single@crerror\eq@false
5044 \let\eq@ampproof@active\eq@false

5045 \eq@define@key{equations,setup}{linesfallback}[true]{%
5046 \eq@decide@select{#3}{#2}{#1}{%
5047   {\eq@decide@false{\let\eq@multi@linesfallback\eq@false}},%
5048   {\reuse,lean}{\let\eq@multi@linesfallback\z@}},%
5049   {\measure,full,\eq@decide@true}{\let\eq@multi@linesfallback\eq@true}}}}
5050 \eq@define@key{setup}{ampproof}[true]{%
5051 \eq@decide@bool{#3}{#2}{#1}\eq@ampproof@active}
5052 \eq@define@key{setup}{crerror}[true]{%
5053 \eq@decide@bool{#3}{#2}{#1}\eq@single@crerror}
5054 \eq@define@key{setup}{modifierwarning}[true]{%
5055 \eq@decide@select{#3}{#2}{#1}{%
5056   {\eq@decide@false}{\let\eq@parseopt@warn\@empty}},%
5057   {\eq@decide@true}{\let\eq@parseopt@warn\eq@warn@parseopt}},%
5058   {\verbose,+}{\let\eq@parseopt@warn\eq@warn@parseopt@verbose}}}}
5059 \let\eq@parseopt@warn\eq@warn@parseopt
5060 \eq@define@key{equations,setup}{rescan}[true]{%
5061 \eq@decide@if{#3}{#2}{#1}%
5062   {\let\eq@scan@body\eq@scan@body@rescan}%
5063   {\let\eq@scan@body\eq@scan@body@dump}}
5064 \eq@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5065 \eq@decide@bool{#3}{#2}{#1}\eq@scan@par}
5066 \eq@define@key{setup}{defaults}{%
5067 \eq@decide@select{#3}{#2}{#1}{%
5068   {\classic}{\eq@defaults@classic}},%
5069   {\eqnlines}{\eq@defaults@eqnlines}}}}

```

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

```

5070 \let\eq@provide@opt@env\tw@
5071 \let\eq@provide@opt@amsmathends\eq@true
5072 \let\eq@provide@opt@backup\eq@false
5073 \let\eq@provide@opt@ang\eq@true
5074 \let\eq@provide@opt@eqref\eq@true

5075 \eq@define@key{setup}{amsmathends}[true]{%
5076 \eq@error@packageoption{#2}%
5077 \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@amsmathends}

```

```

5078 \eql@define@key{setup}{backup}[true]{%
5079   \eql@error@packageoption{#2}%
5080   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
5081 \eql@define@key{setup}{env}[equation]{%
5082   \eql@error@packageoption{#2}%
5083   \eql@decide@select{#3}{#2}{#1}{%
5084     {none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5085     {equation,latex}{\let\eql@provide@opt@env\@ne}},%
5086     {amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}}
5087 \eql@define@key{setup}{ang}[true]{%
5088   \eql@error@packageoption{#2}%
5089   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5090 \eql@define@key{setup}{eqref}[true]{%
5091   \eql@error@packageoption{#2}%
5092   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Shortcut Options. **TODO:** describe

```

5093 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5094 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5095 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5096 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5097 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5098 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5099 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5100 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5101 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
5102 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
5103 \def\eql@parseopt@punctoff#1{\eqnaddopt{punct={}}\eql@parseopt@peek}

```

P.3 Parameter Presets

The package offers two parameter presets which lead to somewhat different layout. Instead of setting the internal parameters directly, we expose them as public settings so that they are easier to read and such that individual settings can be used to compose own layouts.

`\eql@defaults@classic` The preset `classic` aims to reproduce the \TeX , \LaTeX and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

5104 \def\eql@defaults@classic{%
5105   \eqnlineset{numberline=all}%
5106   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5107   \eqnlineset{maxcolsep=off}%
5108   \eqnlineset{spread={\jot}}%
5109   \eqnlineset{tagmargin}%
5110   \eqnlineset{tagmarginratio=1}%
5111   \eqnlineset{tagmarginthreshold=0.5}%
5112   \eqnlineset{leftmargin={\leftmargini}}%
5113   \eqnlineset{padding=max}%
5114   \eqnlineset{evadetag=off}%
5115   \eqnlineset{displayheight=off}%
5116   \eqnlineset{displaydepth=off}%
5117   \eqnlineset{shortmode=belowsingle}%
5118   \eqnlineset{abovecontmode=short}%
5119   \eqnlineset{belowcontmode=short}%
5120   \eqnlineset{aboveparmode=long}%
5121   \eqnlineset{belowparmode=long}%

```

```

5122 \eqnlineset{abovetopmode=long}%
5123 \eqnlineset{belwtopmode=long}%
5124 \eqnlineset{abovelongskip={\abovedisplayskip}}%
5125 \eqnlineset{belowlongskip={\belowdisplayskip}}%
5126 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5127 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5128 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5129 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5130 \eqnlineset{abovecontskip=0pt}%
5131 \eqnlineset{belowcontskip=0pt}%
5132 \eqnlineset{aboveparskip=0pt}%
5133 \eqnlineset{belowparskip=0pt}%
5134 \eqnlineset{abovetopskip=0pt}%
5135 \eqnlineset{belwtopskip=0pt}%
5136 \eqnlineset{abovetagskip=0pt}%
5137 \eqnlineset{belowtagskip=0pt}%
5138 \eqnlineset{crerror=false}%
5139 \eqnlineset{linesfallback=false}%
5140 }

```

values based on 10pt vs 12pt

`\eql@defaults@eqnlines` The (default) preset `eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the classic spacing for a 12 pt computer modern font such that 10 pt fonts will lead to slightly reduced spacing. Apart from that, the `eqnlines` setting makes some deliberate layout choices that deviate significantly from `classic` (maximum column separation, no shortening below equations):

```

5141 \def\eql@defaults@eqnlines{%
5142   \eqnlineset{numberline=all}%
5143   \eqnlineset{mintagsep=.5em}%
5144   \eqnlineset{maxcolsep=2em}%
5145   \eqnlineset{spread={0.2\normalbaselineskip}}%
5146   \eqnlineset{tagmargin}%
5147   \eqnlineset{tagmarginratio=.334}%
5148   \eqnlineset{tagmarginthreshold=0.5}%
5149   \eqnlineset{leftmargin={\leftmargini}}%
5150   \eqnlineset{padding=0pt}%
5151   \eqnlineset{evadetag}%
5152   \eqnlineset{displayheight=strut}%
5153   \eqnlineset{displaydepth=strut}%
5154   \eqnlineset{shortmode=above}%
5155   \eqnlineset{abovecontmode=noskip}%
5156   \eqnlineset{belowcontmode=long}%
5157   \eqnlineset{aboveparmode=long}%
5158   \eqnlineset{belowparmode=long}%
5159   \eqnlineset{abovetopmode=noskip}%
5160   \eqnlineset{belwtopmode=long}%
5161   \eqnlineset{longskip={0.75\normalbaselineskip
5162     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5163   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5164     plus 0.25\normalbaselineskip}}%
5165   \eqnlineset{belowshortskip={0.0\normalbaselineskip
5166     plus 0.25\normalbaselineskip}}%
5167   \eqnlineset{medskip={0.4\normalbaselineskip
5168     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5169   \eqnlineset{abovecontskip=0pt}%
5170   \eqnlineset{belowcontskip=0pt}%

```

```

5171 \eqnlineset{aboveparskip=0pt}%
5172 \eqnlineset{belowparskip=0pt}%
5173 \eqnlineset{abovetopskip=0pt}%
5174 \eqnlineset{belowtopskip=0pt}%
5175 \eqnlineset{abovetagskip={0.25\normalbaselineskip
5176   minus 0.25\normalbaselineskip}}%
5177 \eqnlineset{belowtagskip={0.25\normalbaselineskip
5178   minus 0.25\normalbaselineskip}}%
5179 \eqnlineset{crerror=true}%
5180 \eqnlineset{linesfallback=true}%
5181 }

```

P.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of \LaTeX and `amsmath` carefully.

Tools.

`\eql@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```

\eql@provide@moveenv
\eql@provide@movestart 5182 \def\eql@provide@movecmd#1#2{%
@provide@undefinecmd 5183   \eql@letcs{#1\expandafter}\csname #2\endcsname
@provide@undefineenv 5184 }
5185 \def\eql@provide@moveenv#1#2{%
5186   \eql@provide@movecmd{#1}{#2}%
5187   \eql@markline@amsthm@register{#1}%
5188   \ifcsname end#2\endcsname
5189     \eql@provide@movecmd{end#1}{end#2}%
5190   \fi
5191 }
5192 \def\eql@provide@movestart#1#2{%
5193   \eql@provide@moveenv{#1}{#2}%
5194   \ifcsname #2*\endcsname
5195     \eql@provide@moveenv{#1*}{#2*}%
5196   \fi
5197 }
5198 \def\eql@provide@undefinecmd#1{%
5199   \eql@letcs{#1}\undefined
5200 }
5201 \def\eql@provide@undefineenv#1{%
5202   \eql@provide@undefinecmd{#1}%
5203   \eql@provide@undefinecmd{end#1}%
5204 }

```

Fix Endings for `amsmath` Environments. The `amsmath` derived environments forward their ending routines directly to the ending routines for the main environments `gather`, `multline`, `align`, `aligned`. This causes a problem when the main environments are replaced but the derived ones are still used. We fix the potential problem by copying the ending routines of the main environments to the ending routines of the derived environments.

`\eql@amsmath@endfix` Check whether the original forwarding of an ending routine is still in place (other packages or future updates to `amsmath` might change the behaviour). If so, copy the ending routine into place:


```

5205 \def\eq@amsmath@endfix#1#2{%
5206   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
5207   \expandafter\ifx\csname end#1\endcsname\@tempa
5208     \eq@provide@movecmd{end#1}{end#2}%
5209   \fi
5210 }

```

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

```

5211 \def\eq@amsmath@fixends{%
5212   \eq@amsmath@after{%
5213     \eq@amsmath@endfix{gather*}{gather}%
5214     \eq@amsmath@endfix{multline*}{multline}%
5215     \eq@amsmath@endfix{align*}{align}%
5216     \eq@amsmath@endfix{flalign*}{align}%
5217     \eq@amsmath@endfix{flalign*}{align}%
5218     \eq@amsmath@endfix{alignat}{align}%
5219     \eq@amsmath@endfix{alignat*}{align}%
5220     \eq@amsmath@endfix{xalignat}{align}%
5221     \eq@amsmath@endfix{xalignat*}{align}%
5222     \eq@amsmath@endfix{xxalignat}{align}%
5223     \eq@amsmath@endfix{gathered}{aligned}%
5224     \eq@amsmath@endfix{alignedat}{aligned}%
5225   }
5226 }

```

Backup amsmath Environments. We can backup all amsmath environments *env* to *amsenv* so that they can be used in parallel if needed.

`\eq@provide@backup@amsmath` Copy an amsmath environment *env* to *amsenv* whenever amsmath is loaded: **TODO:** `amsthm`

```

5227 \def\eq@provide@backup@amsmath#1{%
5228   \eq@amsmath@after{%
5229     \eq@provide@moveenv{ams#1}{#1}%
5230     \AddToHook{package/amsthm/after}{\eq@provide@movecmd{ams#1@qed}{#1@qed}}%
5231   }%
5232 }

```

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

```

5233 \def\eq@provide@backup@eqref{%
5234   \eq@amsmath@after{%
5235     \eq@provide@movecmd{amseqref}{eqref}%
5236   }%
5237 }

```

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

```

5238 \def\eq@provide@backup@multlined{%
5239   \AddToHook{package/mathtools/after}{%
5240     \eq@provide@moveenv{amsmultlined}{multlined}%
5241   }%
5242 }

```

`\eq@provide@backup@equation` The \LaTeX environment `equation` is overwritten by several packages to implement their adjustments. Here we cater for adjustments through `amsmath`, `hyperref` and the PDF

tagging mechanism. Copy `equation` and `equation*` whenever `amsmath` is loaded. Whenever `hyperref` is loaded, and `amsmath` is not yet present, backup the original L^AT_EX and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original L^AT_EX `equation`. The PDF tagging mechanism registers `equation` upon `\begin{document}`. We thus need to register all copies of `equation` on our own, so that they can be used with their new names:

```

5243 \def\eql@provide@backup@equation{%
5244   \eql@amsmath@after{%
5245     \eql@provide@moveenv{amseparation}{equation}%
5246     \eql@tagging@register@env{amseparation}%
5247     \eql@provide@moveenv{amseparation*}{equation*}%
5248     \eql@tagging@register@env{amseparation*}%
5249     \AddToHook{package/amsthm/after}{%
5250       \eql@provide@movecmd{amseparation*@qed}{equation*@qed}}%
5251   }%
5252   \AddToHook{package/hyperref/after}{%
5253     \ifpackageloaded{amsmath}{}%
5254     \let\latexequation\H@equation
5255     \let\endlatexequation\H@endequation
5256     \eql@tagging@register@env{latexequation}%
5257     \eql@provide@moveenv{hyperrefequation}{equation}%
5258     \eql@tagging@register@env{hyperrefequation}%
5259     \AddToHook{package/amsthm/after}{%
5260       \eql@provide@movecmd{latexequation@qed}{equation@qed}%
5261       \eql@provide@movecmd{hyperequation@qed}{equation@qed}
5262     }%
5263   }%
5264 }%
5265 \ifpackageloaded{amsmath}{%\ifpackageloaded{hyperref}{}%
5266   \eql@provide@moveenv{latexequation}{equation}%
5267   \eql@tagging@register@env{latexequation}%
5268   \AddToHook{package/amsthm/after}{%
5269     \eql@provide@movecmd{latexequation@qed}{equation@qed}}%
5270 }%
5271 }

```

`@backup@displaymath` **TODO:** describe

```

5272 \def\eql@provide@backup@displaymath{%
5273   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5274   \AddToHook{package/amsthm/after}{%
5275     \eql@provide@movecmd{latexdisplaymath@qed}{displaymath@qed}}%
5276 }

```

`@backup@subequations` The `amsmath` `subequations` environment is adjusted by `hyperref` through an environment hook, but this hook gets applied only later at `\begin{document}`. Hence, we need to supply the hook routine to the new routine ourselves:

```

5277 \def\eql@provide@backup@subequations{%
5278   \eql@amsmath@after{%
5279     \eql@provide@moveenv{amssubequations}{subequations}%
5280   }%
5281   \AddToHook{package/hyperref/after}{%
5282     \AddToHook{cmd/amssubequations/before}{%
5283       {%
5284         \stepcounter{equation}%
5285         \protected@edef\theHparentequation{\theHequation}%
5286         \addtocounter{equation}{-1}%

```

```

5287 }%
5288 \AddToHook{cmd/amssubequations/after}%
5289 {%
5290 \def\theHequation{\theHparentequation\alph{equation}}%
5291 \ignorespaces
5292 }%
5293 }%
5294 }

```

`\eql@provide@backup` Backup all amsmath environments:

```

5295 \def\eql@provide@backup{%
5296 \eql@provide@backup@eqref
5297 \eql@provide@backup@equation
5298 \eql@provide@backup@displaymath
5299 \eql@provide@backup@amsmath{gather}%
5300 \eql@provide@backup@amsmath{gather*}%
5301 \eql@provide@backup@amsmath{multline}%
5302 \eql@provide@backup@amsmath{multline*}%
5303 \eql@provide@backup@amsmath{align}%
5304 \eql@provide@backup@amsmath{align*}%
5305 \eql@provide@backup@amsmath{flalign}%
5306 \eql@provide@backup@amsmath{flalign*}%
5307 \eql@provide@backup@amsmath{alignat}%
5308 \eql@provide@backup@amsmath{alignat*}%
5309 \eql@provide@backup@amsmath{xalignat}%
5310 \eql@provide@backup@amsmath{xalignat*}%
5311 \eql@provide@backup@amsmath{xxalignat}%
5312 \eql@provide@backup@amsmath{aligned}%
5313 \eql@provide@backup@amsmath{aligned*}%
5314 \eql@provide@backup@amsmath{alignedat}%
5315 \eql@provide@backup@amsmath{alignedat*}%
5316 \eql@provide@backup@amsmath{gathered}%
5317 \eql@provide@backup@amsmath{gathered*}%
5318 \eql@provide@backup@multlined
5319 \eql@provide@backup@subequations
5320 }

```

Replacement amsmath Environments. **TODO:** describe

```

5321 \def\eql@alignat@gobblecol#1{%
5322 \eql@ifnextchar@tight\bgroup{\@firstoftwo{#1}}{#1}}

```

`eql@gathered` (*env.*) Define replacement versions for boxed environments `gathered`, `multlined` and `aligned`
`eql@multlined` (*env.*) which forward to `equationsbox` with specific presets:

`eql@aligned` (*env.*)

```

5323 \newenvironment{eql@gathered}
5324 {\eqnadopt{lines}\equationsbox}{\endequationsbox}
5325 \newenvironment{eql@multlined}
5326 {\eqnadopt{lines, padding, shape=steps}\equationsbox}{\endequationsbox}
5327 \newenvironment{eql@aligned}
5328 {\eqnadopt{columns}\equationsbox}{\endequationsbox}
5329 \newenvironment{eql@alignedat}
5330 {\eqnadopt{columns, colsep=off}\eql@alignat@gobblecol\equationsbox}
5331 {\endequationsbox}

```

`eql@equation` (*env.*) Define replacement versions for display environments `equation`, `gather`, `multline`,
`eql@gather` (*env.*) `aligned` and derivatives which forward to `equations` with specific presets: **TODO:**
`eql@multline` (*env.*)
`eql@align` (*env.*)

amsmath at variants would need predefined columns for full operation

```

5332 \newenvironment{eql@equation}
5333   {\eqnaddopt{equation}\equations}\{endequations}
5334 \newenvironment{eql@displaymath}
5335   {\eqnaddopt{equation,nonumber}\equations}\{endequations}
5336 \newenvironment{eql@gather}
5337   {\eqnaddopt{lines}\equations}\{endequations}
5338 \newenvironment{eql@multline}
5339   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5340   {\endequations}
5341 \newenvironment{eql@align}
5342   {\eqnaddopt{columns}\equations}\{endequations}
5343 \newenvironment{eql@flalign}
5344   {\eqnaddopt{fulllength}\eql@align}\{endequations}
5345 \newenvironment{eql@alignat}
5346   {\eqnaddopt{colsep=off}\eql@xalignat}\{endequations}
5347 \newenvironment{eql@xalignat}
5348   {\eql@alignat@gobblecol\eql@align}\{endequations}
5349 \newenvironment{eql@xxalignat}
5350   {\eqnaddopt{fulllength}\eql@xalignat}\{endequations}
5351 \newenvironment{eql@equation*}
5352   {\eqnaddopt{nonumber}\eql@equation}\{endequations}
5353 \newenvironment{eql@gather*}
5354   {\eqnaddopt{nonumber}\eql@gather}\{endequations}
5355 \newenvironment{eql@multline*}
5356   {\eqnaddopt{nonumber}\eql@multline}\{endequations}
5357 \newenvironment{eql@align*}
5358   {\eqnaddopt{nonumber}\eql@align}\{endequations}
5359 \newenvironment{eql@flalign*}
5360   {\eqnaddopt{nonumber}\eql@flalign}\{endequations}
5361 \newenvironment{eql@alignat*}
5362   {\eqnaddopt{nonumber}\eql@alignat}\{endequations}
5363 \newenvironment{eql@xalignat*}
5364   {\eqnaddopt{nonumber}\eql@xalignat}\{endequations}

```

Install Additional Environments. The additional environments need to be installed at their intended names which can be adjusted by the user.

`eql@provide@onlyonce` Process arguments for providing a specific environment. #1 describes the environment using the amsmath name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the amsmath environment in place making sure that the replacement is robust against loading amsmath before or after. If #2 equals '*', just overwrite the amsmath environment in place immediately (e.g. within a block in the document body):

```

5365 \def\eql@provide@onlyonce#1#2{%
5366   \def\eql@tmp{#2}%
5367   \def\@tempa{#1}%
5368   \ifx\eql@tmp\@tempa
5369     \let\eql@tmp\@empty
5370   \fi
5371   \ifx\eql@tmp\@empty
5372     \let\eql@tmp\@undefined
5373   \ifx\@nodocument\relax
5374     \def\eql@tmp{#1}%
5375   \fi
5376   \ifcsname eql@provided@#1\endcsname
5377     \def\eql@tmp{#1}%

```

```

5378   \fi
5379   \eql@letcs{eql@provided@#1}\eql@true
5380 \else
5381   \def\@tempa{*}%
5382   \ifx\eql@tmp\@tempa
5383     \def\eql@tmp{#1}%
5384   \fi
5385 \fi
5386 }

```

`\eql@provide@eqref` Provide `\eqref` as the macro #1. We have to check whether #1 is empty or equals `\eqref` or takes the value `*`. If not, we should strip the backslash for further processing. Copy the macro into place, and copy again when `amsmath` or `mathtools` are loaded. Remove definition before `amsmath` is loaded in the future to avoid a potential error:

```

5387 \def\eql@provide@eqref#1{%
5388   \def\eql@tmp{#1}%
5389   \def\@tempa{\eqref}%
5390   \ifx\eql@tmp\@tempa
5391     \let\eql@tmp\@empty
5392   \fi
5393   \ifx\eql@tmp\@empty
5394     \eql@provide@onlyonce{eqref}{}%
5395   \else
5396     \def\@tempa{*}%
5397     \ifx\eql@tmp\@tempa
5398       \def\eql@tmp{eqref}%
5399     \else
5400       \edef\eql@tmp{\expandafter\@gobble\string#1}%
5401     \fi
5402   \fi
5403   \ifdefined\eql@tmp
5404     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
5405   \else
5406     \eql@amsmath@after{%
5407       \eql@provide@movecmd{eqref}{eql@eqref}%
5408     }%
5409     \AddToHook{package/mathtools/after}{%
5410       \eql@provide@movecmd{eqref}{eql@eqref}%
5411     }%
5412     \eql@provide@movecmd{eqref}{eql@eqref}%
5413     \eql@amsmath@undefine\eqref
5414   \fi
5415 }

```

`\eql@provide@amsmath` Provide one of the `amsmath` environments and its star variant. Copy into place, and copy again when `amsmath` or `mathtools` are loaded. Remove definition before `amsmath` is loaded in the future to avoid an error:

```

5416 \def\eql@provide@amsmath#1#2{%
5417   \eql@provide@onlyonce{#1}{#2}%
5418   \ifdefined\eql@tmp
5419     \expandafter\eql@provide@movestar\expandafter{\eql@tmp}{eql@#1}%
5420   \else
5421     \eql@amsmath@after{%
5422       \eql@provide@movestar{#1}{eql@#1}%
5423     }%
5424     \AddToHook{package/mathtools/after}{%

```

```

5425     \eql@provide@movestarm{#1}{eql@#1}%
5426   }%
5427   \eql@provide@movestarm{#1}{eql@#1}%
5428   \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
5429   \ifcsname eql@#1*\endcsname
5430     \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
5431   \fi
5432 \fi
5433 }

```

`\eql@provide@multlined` Provide `mathtools` environment `multlined`. Copy into place, and copy again when `amsmath` or `mathtools` are loaded. Remove definition before `mathtools` is loaded in the future to avoid an error:

```

5434 \def\eql@provide@multlined#1{%
5435   \eql@provide@onlyonce{multlined}{#1}%
5436   \ifdefined\eql@tmp
5437     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
5438   \else
5439     \AddToHook{package/mathtools/after}{%
5440       \eql@provide@moveenv{multlined}{eql@multlined}%
5441     }%
5442     \eql@provide@moveenv{multlined}{eql@multlined}%
5443     \@ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5444       \eql@provide@undefineenv{multlined}}}%
5445   \fi
5446 }

```

`\eql@provide@equation` Provide the environment `equation` and its star variant. Copy into place, and copy again when `amsmath` or `hyperref` are loaded. Remove definition of `equation*` before `amsmath` is loaded in the future to avoid an error. When PDF tagging is active, the environment is modified at `\begin{document}` in an undesirable fashion, so copy the definition again:

```

5447 \def\eql@provide@equation#1{%
5448   \eql@provide@onlyonce{equation}{#1}%
5449   \ifdefined\eql@tmp
5450     \expandafter\eql@provide@movestarm\expandafter{\eql@tmp}{eql@equation}%
5451   \else
5452     \eql@amsmath@after{%
5453       \eql@provide@movestarm{equation}{eql@equation}%
5454     }%
5455     \AddToHook{package/hyperref/after}{%
5456       \@ifpackageloaded{amsmath}{-%
5457         \eql@provide@moveenv{equation}{eql@equation}%
5458       }%
5459     }%
5460     \eql@provide@movestarm{equation}{eql@equation}%
5461     \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5462     \ifdefined\eql@tagging@on
5463       \AddToHook{begindocument/end}{%
5464         \eql@provide@movestarm{equation}{eql@equation}%
5465       }%
5466     \fi
5467   \fi
5468 }

```

`\eql@provide@displaymath` **TODO:** describe

```

5469 \def\eql@provide@displaymath#1{%
5470   \eql@provide@onlyonce{displaymath}{#1}%
5471   \ifdefined\eql@tmp
5472     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@displaymath}%
5473   \else
5474     \eql@provide@moveenv{displaymath}{\eql@displaymath}%
5475     \ifdefined\eql@tagging@on
5476       \AddToHook{begindocument/end}{%
5477         \eql@provide@moveenv{displaymath}{\eql@displaymath}%
5478       }%
5479     \fi
5480   \fi
5481 }

```

`\provide@subequations` Provide the amsmath environment `subequations`. Copy into place, and copy again when `amsmath` is loaded. `hyperref` adds a hook to the command which messes up the parsing of optional arguments (even if the hook is emptied). The hook placement happens at `\begin{document}`, so we copy the environment again afterwards. We also remove the hook (after adding an empty hook to avoid errors). Remove definition before `amsmath` is loaded in the future to avoid an error:

```

5482 \def\eql@provide@subequations#1{%
5483   \eql@provide@onlyonce{subequations}{#1}%
5484   \ifdefined\eql@tmp
5485     \expandafter\eql@provide@moveenv
5486     \expandafter{\eql@tmp}{\eql@subequations}%
5487   \else
5488     \eql@amsmath@after{%
5489       \eql@provide@moveenv{subequations}{\eql@subequations}%
5490     }%
5491     \AddToHook{package/hyperref/after}{%
5492       \AddToHook{cmd/subequations/before}[hyperref]{}%
5493       \AddToHook{cmd/subequations/after}[hyperref]{}%
5494       \RemoveFromHook{cmd/subequations/before}[hyperref]%
5495       \RemoveFromHook{cmd/subequations/after}[hyperref]%
5496       \AddToHook{begindocument/end}{%
5497         \eql@provide@moveenv{subequations}{\eql@subequations}%
5498       }%
5499     }%
5500     \eql@provide@moveenv{subequations}{\eql@subequations}%
5501     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
5502   \fi
5503 }

```

`\eql@provide@sqr` Provide the symbolic environment `\[...]`. Copy into place, and copy again when `amsmath` is loaded. If PDF tagging is active, some undesired modifications happen at `\begin{document}`, so copy again afterwards:

```

5504 \def\eql@provide@sqr{%
5505   \let\[ \eql@sqr@open
5506   \let\] \eql@sqr@close
5507   \eql@amsmath@after{%
5508     \let\[ \eql@sqr@open
5509     \let\] \eql@sqr@close
5510   }%
5511   \ifdefined\eql@tagging@on
5512     \AddToHook{begindocument/end}{%
5513       \let\[ \eql@sqr@open

```

```

5514     \let\}\eql@sqr@close
5515   }%
5516   \fi
5517 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...>`. This is easy because none of the other packages uses this structure:

```

5518 \def\eql@provide@ang{%
5519   \let\<\eql@ang@open
5520   \let\>\eql@ang@close
5521 }

```

Interface.

`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

5522 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5523 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5524 \eql@define@key{provide}{gather}[]{\eql@provide@amsmath{gather}{#1}}
5525 \eql@define@key{provide}{multline}[]{\eql@provide@amsmath{multline}{#1}}
5526 \eql@define@key{provide}{align}[]{\eql@provide@amsmath{align}{#1}}
5527 \eql@define@key{provide}{flalign}[]{\eql@provide@amsmath{flalign}{#1}}
5528 \eql@define@key{provide}{alignat}[]{\eql@provide@amsmath{alignat}{#1}}
5529 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsmath{xalignat}{#1}}
5530 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsmath{xxalignat}{#1}}
5531 \eql@define@key{provide}{aligned}[]{\eql@provide@amsmath{aligned}{#1}}
5532 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsmath{alignedat}{#1}}
5533 \eql@define@key{provide}{gathered}[]{\eql@provide@amsmath{gathered}{#1}}
5534 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5535 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5536 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
5537 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
5538 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
5539 \eql@define@key{provide}{tagform}[]{%
5540   \def\tagform@##1{\maketag@@{\eql@tags@tagform{#1}}}}
5541 \eql@define@key{provide}{maketag}[]{%
5542   \def\maketag@@@##1{\eql@tags@taglayout{#1}}}

```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```

5543 \newcommand{\eqnlinesprovide}[1]{%
5544 (dev)\eql@dev@start\eqnlinesprovide
5545   \eql@setkeys{provide}{#1}%
5546   \ignorespaces
5547 }

```

P.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
5548 \let\eql@error@packageoption@gobble
```

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


```
5549 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
5550 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
5551 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
5552 \eql@defaults@eqnlines
5553 \eql@mode@columns
5554 \eql@mode@aligned
```

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

```
5555 \ifdefined\tagsleft@true\else
5556   \expandafter\newif\csname iftagsleft@\endcsname
5557 \fi
5558 \ifdefined\@fleqntrue\else
5559   \expandafter\newif\csname if@fleqn\endcsname
5560 \fi
```

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

```
5561 \eql@amsmath@after{%
5562   \ifnum\eql@provide@opt@env=\tw@
5563     \iftagsleft@
5564       \eqnlineset{tags=left}%
5565     \else
5566       \eqnlineset{tags=right}%
5567     \fi
5568   \if@fleqn
5569     \eqnlineset{layout=left}%
5570   \else
5571     \eqnlineset{layout=center}%
5572   \fi
5573 \fi
5574 }
```

Process package options:

```
5575 \ProcessOptions
```

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

```
5576 \def\eql@error@packageoption#1{%
5577   \eql@error{may only use '#1' as a package option}%
5578 }
```

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

```
5579 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

Backup all `amsmath` environments that may be overwritten to `ams...`. This will happen before any replacements:

```
5580 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

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

```
5581 \ifnum\eql@provide@opt@env>\z@
```

```
5582 \eqnlinesprovide{equation,sqr,displaymath}
5583 \fi
```

Provide amsmath equation environments if desired:

```
5584 \ifnum\eq1@provide@opt@env=\tw@
5585 \eqnlinesprovide{%
5586     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
5587     multlined,gathered,aligned,alignedat,%
5588     subequations}
5589 \fi
```

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

```
5590 \ifdefined\eq1@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
5591 \ifdefined\eq1@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```