

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`

2025/05/29, v0.10

<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	11
2.4	Punctuation	16
2.5	Math Classes at Alignment	17
2.6	Vertical Spacing	18
2.7	Further Environments and Features	21
2.8	General Options	23
2.9	Feature Selection and Package Options	24
3	Information	24
3.1	Copyright	24
3.2	Credits	25
3.3	Files and Installation	25
3.4	Related CTAN Packages	26
3.5	Feature Suggestions	26
3.6	Revision History	27
A	Implementation	28

B	General Support	29
B.1	Development Messages	29
B.2	Supporting Definitions	29
B.3	Dollardollar Abstraction	29
B.4	Look-Ahead in Alignment	29
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	42
D	Features	43
D.1	Punctuation	43
D.2	Math Classes at Alignment	44
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	50
E.4	Integration	55
E.5	Positioning	58
E.6	Component Display	62
E.7	Tag Composition	65
F	Subequation Numbering	67
F.1	Definitions	67
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	79
H	Multi-Line Support	81
H.1	Measure Support	81
H.2	Line Breaks	82
H.3	Intertext	83
H.4	Line Marks	85
I	Column Placement	87
I.1	Supporting Definitions	87
I.2	Shape Schemes	87
I.3	Width Data	90
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	108
K.1	Line Breaks	109
K.2	Stacked Mode	109
K.3	Aligned Mode	110
K.4	Main	112
K.5	Environment	113
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	128
O.1	Scanning the Equation Body	128
O.2	Options Processing	132
O.3	Single-Line Main	133
O.4	Multi-Line Main	134
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	151
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: Please note that this package is still in a development and testing stage in the present version. This mainly applies to the documentation

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

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

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

To use the `eqnlines` package add the command

$$\backslash usepackage{eqnlines}$$

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

2.1 Equations Environment

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

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

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

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

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

$$\backslash eqnlineset{opts}$$

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

$$\backslash eqncontrol{opts}$$

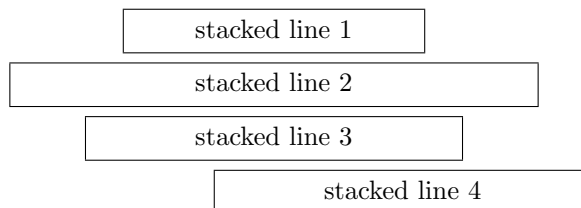
The `\eqncontrol` interface also provides several features for which no other macro definitions exist. Shortcuts to frequently used features could be installed by user definitions such as:

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

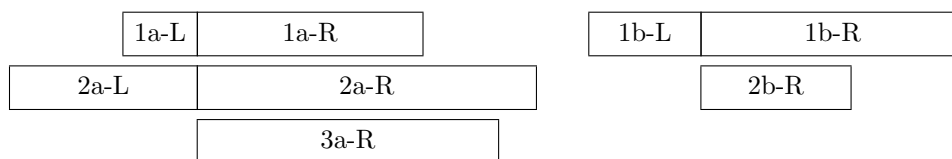
Modes of Operation. The package supplies a main maths environment called `equations` which has three principal modes of operation. It can display a single-line equation just as the \LaTeX environment `equation` or the symbolic shortcut `\[...\]`:

single line

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	multiple, aligned
alignment	adjustable	adjustable	alternating right/left
parsing	single, direct	two passes	two passes
numbering	on/off	off/single/multiple	off/single/multiple

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

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

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.

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

<code>\[</code>			
<code>x=\cos\phi</code>		$x = \cos \phi$	
<code>\]</code>			
<code>\<=</code>			
<code>x=\cos\phi \\\phi=\arccos x</code>		$x = \cos \phi$	
<code>\></code>		$\phi = \arccos x$	
<code>\<</code>			
<code>x&=\cos\phi & \phi&=\arccos x \\\&=(z+z^{-1})/2 & \&=-i\log z</code>	$x = \cos \phi$	$\phi = \arccos x$	
<code>\></code>		$= (z + z^{-1})/2$	$= -i \log z$
<code>\eqnlineset{sqropt={donumber}}</code>			
<code>\[x=\cos\phi \]</code>		$x = \cos \phi$	(6)

`equation` (env.) The package also supplies or overwrites the `amsmath` environments `equation`, `gather`, `gather` (env.) `multline`, `align` and `flalign` including their starred at -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}`

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

`transpose` (key) **Transposition.** When the aligned mode is used to produce more than one column of equations, the default line-by-line ordering of the content may be inconvenient. The package offers a transposition mode `transpose=plain` in which the content is specified on a column-by-column basis. Columns are separated by ‘`\&`’ (the character ‘`&`’ must be escaped as ‘`\&`’ in this mode) and the lines within each column are broken by ‘`\`’ as usual. The continued

transposition mode `transpose=cont` (abbreviated by the modifier `/`) furthermore reduces the input by assuming that all secondary alignment markers `&` indicate a continued equation and imply a line break with an empty left equation cell. Note that the transposition is implemented by reprocessing the input, which imposes some restrictions: all line and column breaks `\`, `\&` must be explicit (must not be produced by macro expansion), line breaks should not use optional arguments (they only work on the first column), and each section separated by `\&` should describe only a single column with one alignment marker per line (unless in continued transposition mode). Furthermore, the continued mode processes the alignment marker `&`, which may cause issues when nesting aligned content.

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

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

2.2 Numbering

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

```
\begin{equations}[!,numberline=...]
  x  &= \cos\phi  \ \ \&= (z+z^{-1})/2 \ \ \
\phi &= \arccos x \ \ \&= -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:	
$x = - \int \sin \phi \, d\phi$		$x = - \int \sin \phi \, d\phi$		$x = - \int \sin \phi \, d\phi$	
$= \cos \phi$	(24)	$= \cos \phi$	(25)	$= \cos \phi$	(26)

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

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

\nonumber and **\donumber**

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

nn,* (*key*)

dn,! (*key*)

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

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

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

\numbernext

\numberhere and **\numbernext**

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

<code>\begin{equations}</code>	
<code> x &= \cos\phi \nonumber \\</code>	$x = \cos \phi$
<code> &= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &= \arccos x \nonumber \\</code>	$\phi = \arccos x$
<code> &= -i\log z</code>	$= -i \log z$
<code>\end{equations}</code>	(28)
<code>\begin{equations}*</code>	
<code> x &= \cos\phi \donumber \\</code>	$x = \cos \phi$
<code> &= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &= \arccos x \donumber \\</code>	$\phi = \arccos x$
<code> &= -i\log z</code>	$= -i \log z$
<code>\end{equations}</code>	(29)
<code>\eqnlineset{numberline=last}</code>	
<code>\<! x &= \cos\phi \\</code>	$x = \cos \phi$
<code>\phi &= \arccos x \></code>	$\phi = \arccos x$
<code>\eqnlineset{angopt=donumber}</code>	
<code>\<* x &= \cos\phi \\</code>	$x = \cos \phi$
<code>\phi &= \arccos x \></code>	$\phi = \arccos x$
<code>\begin{equations}</code>	
<code> x &= \cos\phi \numbernext \\</code>	$x = \cos \phi$
<code> &= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &= \arccos x \numbernext \\</code>	$\phi = \arccos x$
<code> &= -i\log z</code>	$= -i \log z$
<code>\end{equations}</code>	(32)
<code>\eqnlineset{numberline=here}</code>	
<code>\<! x &= \cos\phi \\</code>	$x = \cos \phi$
<code> &= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &= \arccos x \numberhere \\</code>	$\phi = \arccos x$
<code> &= -i\log z</code>	$= -i \log z$
<code>\></code>	(34)
<code>\eqnlineset{numberline=first}</code>	
<code>\<! x &= \cos\phi \numbernext \\</code>	$x = \cos \phi$
<code> &= (z+z^{-1})/2 \\</code>	$= (z + z^{-1})/2$
<code>\phi &= \arccos x \numbernext \\</code>	$\phi = \arccos x$
<code> &= -i\log z</code>	$= -i \log z$
<code>\></code>	(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
`\tagbox` decorated number in a box and `\tagboxed` combines the two.
`\tagboxed` The typesetting of equation numbers and tags passes through two macros, one which defines
`tagbox (key)` the layout and another one which adds a decoration by parentheses. These two methods
`tagform (key)` can be adjusted via the options:

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

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

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

```

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

```

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

[36]

2.3 Horizontal Placement

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

`left (key)`

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

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

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

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

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

tagmargin (*key*) In central alignment layout, one can impose a tag margin **tagmargin={dimen}** which allocates some space to the tag such that equation content is centred in the remaining horizontal space. The margin can also be set to the width of some text by **tagmargin*={text}** or it can be calculated as the maximum width of tags by **tagmargin** without parameter (default). The option **tagmarginratio={ratio}** uses the tag margin only for equation blocks with a ratio of tags to rows above the given (decimal) ratio (a value above 1 uses the tag margin only for single equations with tags; default is 0.334). The option **tagmarginthreshold={threshold}**

uses the tag margin only if the ratio of spacings would be below the given (decimal) threshold (very much off balance; default is 0.5). The latter two options together with some tag margin can produce a more appealing layout for equation blocks of mixed filling. In the following example, the former two equations are centred on all horizontal space while the latter two equations are centred on the space left of the tag (the ratio of spacings without tag margin would be very small here):

```

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

```

(45)

(46)

(47)

(48)

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

```

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

```

(49) $x = \cos \phi$

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

(51) $\phi = \arccos x$

(52) $= -i \log z$

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

```

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

```

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

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

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

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

name	alt.	shape	alignment
default	def	uniform	default
left	l		left
center	c	uniform	central
right	r		right
first	indent, rc	first/rest	first line indented
hanging	outdent, lc	first/rest	first line hanging
steps	lcr	first/intermediate/last	left/centre...centre/right

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

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

`shiffto (key)`

`shiftby (key)`

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

`\eqncontrol{shiffto|shiftby=dimen}`

`\shoveleft` or by the macros:

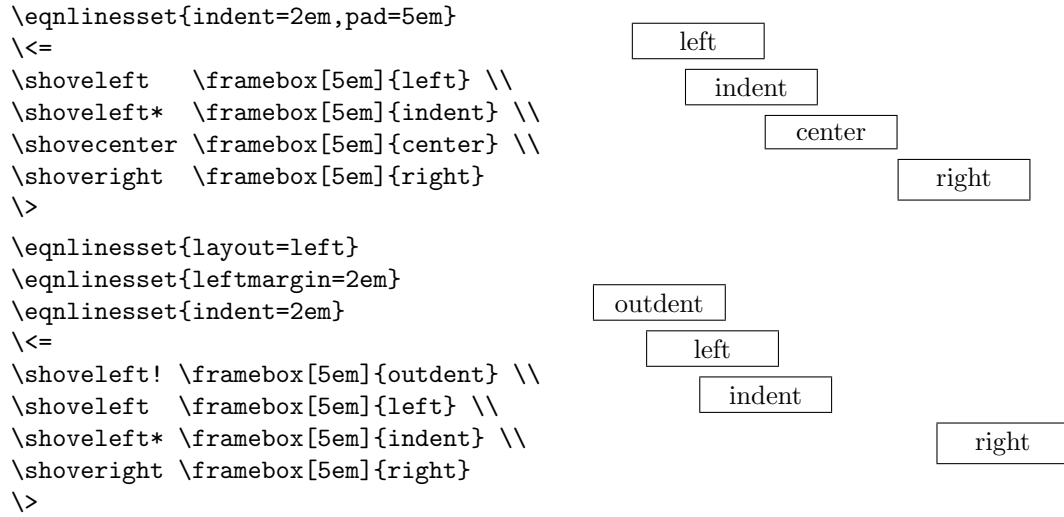
`\shovecenter`

`\shoveright`

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

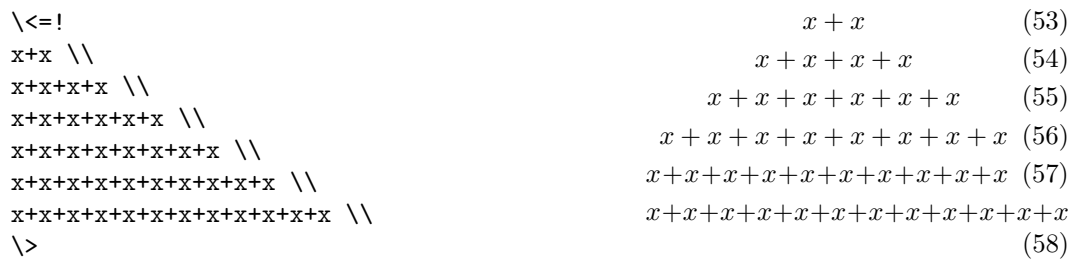
In contradistinction to `amsmath`, these macros can be placed anywhere within the cell and they do not take the cell contents as their argument (doing this here will disallow shrinking of glue towards reducing width). The macros accept an optional argument `[dimen]` specifying a variable amount of shift.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

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



Fitting. Finally, we note that the package will make attempts at fitting the equation components into the horizontal space by adjusting some dimensions with the priority of avoiding overlong lines. The adjustments will first concern the intercolumn and margin spacing. Secondly, T_EX will attempt to shrink the glue between symbols for very wide single and stacked equations (but not aligned equations). Finally, equation tags may be shifted out of the way vertically in order to free up horizontal space. If all attempts fail, overlong lines will be indicated.

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



mintagsep (*key*) If the available space on a line does not suffice to place both the equation and its tag (with a minimum separation of `mintagsep`; default is 0.5em), a tag will automatically be shifted

(lowered or raised depending on whether it is placed on the right or left) to an otherwise empty line. The `\eqncontrol` control `shifttag=dimen` (alternatively `\raisetag*`) may be used to shift a tag up (or down with negative arguments). The control `smashtag=dimen` (alternatively `\raisetag`) may be used to fine-tune the vertical placement when the tag requires extra vertical space but some space above or below the tag is unoccupied. It smashes some of the tag's height (or depth with negative arguments) and thus reduces the vertical gap created by the tag. Note that this feature can be used successively with positive and negative arguments to reduce the space in both directions if available. Where needed, the control `pushtag` (or `\raisetag!`) force-pushes the tag to a separate line and frees up the horizontal space occupied by the tag. The numbering modes `top`, `bottom`, `center`, `median`, `center!` and `center*` are special in that they allow for a continuous vertical placement of the tag between two lines. The more flexible placement of tags may also be enabled for the single-lines modes by the option `tagbetween`. Here, both lines must have sufficiently much space available for the tag. If not, the tag is shifted up or down or it is placed on separate line between the two. The option `tagsnap` defines a range within which the tag baseline snaps to a nearby math baseline.

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

2.4 Punctuation

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

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

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

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

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.

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

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

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

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

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

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

2.5 Math Classes at Alignment

Alignment in multi-line equations breaks equations into components before and after the alignment position. Unfortunately, this also interrupts T_EX’s math spacing mechanism which is based on the math classes assigned to the characters, and there appears to be no direct way of determining the math class to the previous letter. Therefore, one has to make some assumptions on the letters that will surround the alignment marker ‘&’ in order to obtain the appropriate spacing also across the alignment.

The `amsmath` environment `align` assumes that the left column ends with an ordinary character. This leads to the correct spacing when an equation $a = b + c$ is broken before the equals relation as `a&b+c`, and also if an equation sequence continues on the next line as `\\&d-e`. However, it is difficult to achieve the right spacing if the right-hand side is to be broken into several lines: For instance, `\\&_+f` aligns the subordinate binary operation with the equals sign (which may be undesirable). Instead placing a phantom equals sign is an effort that somewhat disrupts the readability of the code.

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

```

$$a = b + c$$

$$= d - e$$

$$+ f$$

`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

`displayheight` (*key*) text. The apparent height and depth are defined by `displayheight` and `displaydepth`
`displaydepth` (*key*) which default to the dimensions of a strut.

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

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

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

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

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

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

There are three special situations `cont`, `par` and `top` which trigger different spacings: `cont` describes the situation at the start of an empty horizontal list (invoked by `\noindent`) or when an equation block directly follows another one; here, the space above the equation should be minimal (or even negative to remove the space below the previous equation block). `par` describes the situation at the beginning of a paragraph (invoked by `\par`); here, the space above the equation adds to the space between paragraphs. `top` describes the situation at the top of a vertical list (invoked by `\nointerlineskip`); here, one would typically want no space.

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

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

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

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

`par` (*key*) 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 as follows: The setting `allowbreaks` (or `allowdisplaybreaks`) taking values 0 (never) 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*) `interpenalty` (*key*) `\displaybreak`

4 (enforce) suggest a break just above or below the equation environment. The command `\displaybreak[val]` with values 0 through 4 (default) suggests a break below the current 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 vertical alignment of the box. `margin`, `marginleft`, `marginright` specify additional margin space around the equations box. `colsep` specifies the amount of separation between the columns. `frame[=cmd]` encloses the equations box by a *cmd* such as `\fbox` which accepts one argument (or a command sequence which ends with a macro accepting one argument). `marginleft` (*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]`

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

`\end{equationsbox}`

`\right\}`

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

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

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

`\>\Longleftarrow$`

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:

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

`\eqnlineset`

`{subeqtemplate={#1-\roman{#2}}}`

`\begin{subequations}`

`\[! x = \cos \phi \]` and $x = \cos \phi$ (62-i)

and $\phi = \arccos x$ (62-ii)

`\[! \phi = \arccos x \]`

`\end{subequations}`

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

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

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 `\eqncontrol{inject*={cmd}}` injects after interline spacing instead.

<code>\< x &= \cos\phi</code>		$x = \cos \phi$
<code>\eqncontrol{inject=\hrule} \\\</code>	<hr style="width: 50%; margin: 0 auto;"/>	
<code>\phi &= \arccos x \></code>		$\phi = \arccos x$

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:

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

<code>\<[n=1]! x &= \cos\phi</code>		$x = \cos \phi$
<code>\eqncontrol{markline={symbol=\sqrt{}}}\ \\\</code>		$\phi = \arccos x$
<code>\phi &= \arccos x</code>		$\phi = \arccos x$
<code>\eqncontrol{qed={shift=.5ex}} \></code>		$\phi = \arccos x$ <div style="text-align: right;"> $\sqrt{}$ (63) QED </div>

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:

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

`alt` (*key*) **Alternative Content Description.** The package provides a basic interface to describe 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}]</code>		
<code>x &= \cos\phi \\\</code>	$x = \cos \phi$	
<code>\eqnalt[line]{reverse relationship}</code>	$\phi = \arccos x$	
<code>\phi &= \arccos x \></code>		

2.8 General Options

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

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

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

The package supplies the following general settings:

option	description
<code>defaults=classic</code>	mimic classic L ^A T _E X/amsmath (layout and dimensions)
<code>defaults=eqnlines</code>	eqnlines layout with fontsize-relative dimensions
<code>rescan</code>	rescan environment body for special commands (e.g. <code>\verb</code>)
<code>linesfallback</code>	single column in align mode reverts to lines mode
	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)LaTeX on `eqnlines.dtx` to compile the manual `eqnlines.pdf` (this file).
- Run LaTeX on `eqnlines.ins` to create the package `eqnlines.sty` and the developers version `eqnlines-dev.sty`. Copy the file `eqnlines.sty` to an appropriate directory of your LaTeX distribution, e.g. `texmf-root/tex/latex/eqnlines`.

3.4 Related CTAN Packages

The package is related to other packages available at CTAN:

- This package uses the package `keyval` to process the options for the package, environments and macros. Compatibility with the `keyval` package has been tested with v1.15 (2022/05/29).
- This package reproduces the math environments functionality of the package `amsmath`. The present code is based on `amsmath` v2.17t (2024/11/05). Compatibility with the `amsmath` package is maintained whether `eqnlines` is loaded before or after `amsmath`. By default, `eqnlines` overwrites most math environments of `amsmath` with its own implementations. It can also preserve them as `ams...` if needed. Alternatively, `eqnlines` may assign individual names to the maths environments and preserve the ones of `amsmath`. The other features provided by `amsmath` can be used.
- The package `mathtools` is a popular extension of the `amsmath` package. This package incorporates some of the features and improvements provided by the `mathtools` package. Compatibility with the `mathtools` package has been tested with v1.31 (2024/10/04), and it is maintained whether `eqnlines` is loaded before or after `mathtools`. Some features like emphasising equations via `empheq` do not (yet) work.
- This package cooperates with the package `hyperref` to create anchors and references within the electronic document. Compatibility with the `hyperref` package has been tested with v7.01l (2024/11/05).
- This package 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: 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 \@currenvir}}
6 <dev>\def\eql@dev@leaveenv{\eql@dev{ leaving \@currenvir}}
7 <dev>\def\eql@dev@in#1#2{\eql@dev{ \space within \string#1 #2}}
```

B.2 Supporting Definitions

`\eql@false` (*bool*) Rather than the standard L^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@dollar@begin` As of 2025 L^AT_EX defines `\dollar@dollar@begin` and `\dollar@dollar@end` to represent (and adjust) the beginning and end of bare T_EX display equations (`‘$$$’`). For the time being, we make sure to revert to `‘$$$’` if these macros are not yet available:

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

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

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

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

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

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

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

```

58 \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}
59 \long\def\eq@gobbleopt[#1]{}
60 \long\def\eq@gobbleoptone[#1]#2{}

```

TODO: describe

```

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

```

TODO: describe

```

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

```

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

```

81 \def\eq@spbgroup{\iffalse{\fi\ifnum0=‘}\fi}
82 \def\eq@speggroup{\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` 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}

```

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>.

```

94 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
95 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
96 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
97 \long\def\eql@teststaropt@tight@ampsafe{%
98   \eql@ampprotecttwo\eql@teststaropt@tight}

```

`\eql@amproof` We may want to replace L^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 `\eql@amproof` overwrites the original definitions, and `\eql@amprevert` reverts the changes:

```

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

```

B.5 Error Messages

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

```

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

```

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

```

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

```

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

```

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

```



```

135   {tag label declaration}{}}

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

```

B.6 amsmath Integration

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

```

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

`\eql@tagging@...` Proper PDF tagging⁴ support requires a \LaTeX (development) version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with \LaTeX versions around 2020: **TODO:** adjust to further developments

```

148 \let\eql@tagging@on\eql@false
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@tablesaveinner{%

```

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

```

170 \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
171 \def\eql@tagging@tablesavealign{%
172 \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
173 \def\eql@tagging@alignleft{%
174 \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
175 \def\eql@tagging@aligncenter{%
176 \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
177 \def\eql@tagging@alignright{%
178 \UseTaggingSocket{math/luamml/mtable/aligncol}{right}}

```

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

```

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

```

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

```

181 \def\eql@tagging@start{%
182 \csname bool_if:N\expandafter\endcsname
183 \csname l__math_collected_bool\endcsname{%
184 \toks@{\expandafter{\eql@tagging@opt}}%
185 \edef\eql@tmp{{\@currentvir}{\the\toks@}\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@tablesavelines{}
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.

`\eqld@decide@select` Some parameter values take values in a given set, e.g. `true` vs. `false` or `left` vs. `right`. The macro `\eqld@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\eqld@decide@relax{\@tempb:=\relax}
214 \def\eqld@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\eqld@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     \eqld@error{undefined value '#3' for option '#2' of '#1'}%
235   \fi
236 }
```

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

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

`\eqld@decide@if`

```
239 \def\eqld@decide@if#1#2#3#4#5{%
240   \eqld@decide@select{#1}{#2}{#3}{%
241     {\eqld@decide@true{#4}},%
242     {\eqld@decide@false{#5}}}%
243 }
```

`\eqld@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@}

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\eq@skip@mode@short` The setting `\eq@skip@mode@short` specifies when a reduced amount of glue should be used around equations in case the text line above the equation fits in the space that is left

available in the first equation line. Value 0 turns this feature off, value 1 reduces the glue above the equation, value 2 furthermore reduces the glue below a single equation line and value 3 also reduces the glue below multi-line equations:

```

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

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

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

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

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

```

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

```

\eq@skip@force@below
\mode@above@ (counter) 332 \newcount\eq@skip@mode@above@
\mode@below@ (counter) 333 \newcount\eq@skip@mode@below@
334 \let\eq@skip@force@above@\@undefined
335 \let\eq@skip@force@below@\@undefined
336 \let\eq@skip@custom@above@\@undefined
337 \let\eq@skip@custom@below@\@undefined

```

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

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

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

`\eq@skip@med@above` The surrounding glue for an equation with reduced spacing is given by `\eq@skip@med@above` and `\eq@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\eql@colseppmin@
353 \def\eql@colseppmin@val{1em}
354 \def\eql@colseppmax@val{.5\maxdimen}

```

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

```

355 \newdimen\eql@tagwidthmin@
356 \eql@tagwidthmin@ \z@

```

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

```

357 \newdimen\eql@tagseppmin@
358 \def\eql@tagseppmin@val{.5\fontdimen6\textfont\tw@}

```

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

```

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

```

Multi-Line Align Mode.

```

362 \let\eql@columns@fulllength\eql@false

```

C.2 Registers

TODO: describe

General. **TODO:** describe

```

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

```

TODO: describe

```

366 \let\eql@display@container\@empty

```

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

```

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

```

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

\@totalwidth@ (*dimen*)
 \@tagwidth@max@ (*dimen*)
 \@totalheight@ (*dimen*)

```

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

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

```

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

\@line@width@ (*dimen*)
 \@line@avail@ (*dimen*)
 \@line@pos@ (*dimen*)
 \@widthsep@ (*counter*)
 \@availsep@ (*counter*)
 \@possep@ (*counter*)
 \@line@offset@ (*dimen*)
 \@prevdepth@ (*dimen*)
 \@interline@ (*dimen*)

```

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

Rows and Columns.

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

\@tagrows@ (*counter*)

```

390 \newcount\eql@row@
391 \newcount\eql@totalrows@
392 \newcount\eql@tagrows@
    
```

\@column@
 \@totalcolumns@

```

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

\@colsep@ (*dimen*) The dimension of the intercolumn separation for align environments is stored in
 \@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
 $\text{\@display@firstavail@set}$ $\text{\@display@firstavail@}$ for comparison against \predisplaysize and determining
 short skip mode of display equations. It is convenient to set it via
 $\text{\@display@firstavail@set}$ provided that we are on the first line:

```

397 \newdimen\eql@display@firstavail@
    
```

```

398 \def\eqldisplay@firstavail@set#1{%
399   \ifnum\eql@row@=\@ne
400     \global\eql@appendexpand\eqldisplay@container{%
401       \eqldisplay@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\eqldisplay@aboveextend@
405 \newdimen\eqldisplay@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 punctuation for columns, lines and blocks. Note that the macro `\eqnpunct` sets the punctuation for the following equation block or for the current cell. Starred versions clear the punctuation for the respectively levels:

```

436 \def\eqnpunct{%
437   \eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
438 \def\eql@punct@next@set#1{%
439   \ifmmode
440     \def\eql@punct@col{#1}%
441     \def\eql@punct@line{#1}%
442     \def\eql@punct@block{#1}%
443   \else
444     \eqnaddopt{punct={#1}}%
445   \fi
446   \ignorespaces}
447 \def\eql@punct@next@setrelax{%
448   \ifmmode
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 `\eq@punct@apply@col` prevent further output of punctuations for lines and columns within the current group:

`\eq@punct@apply@line`

```

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

```

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

```

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

486 \let\eqnpunctapply\eq@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 `\eq@class@innerlead` and `\eq@class@innerright` depending on whether the left part of the aligned column is empty:

```

487 \def\eq@class@innerright@sel{%
488   \ifdim\eq@prevwidth=\z@
489     \eq@class@innerlead
490   \else
491     \eq@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
```

`\eql@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@\@ne}
639 \def\eql@numbering@init@last{\eql@tagpos@row@\@MM}
640 \def\eql@numbering@init@here{\eql@tagpos@row@\m@ne}

```

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

```

TODO: describe

```

655 \def\eql@tagpos@eval@middle{%
656   \ifnum\eql@tagpos@row@=\z@
657     \eql@tagpos@row@\numexpr(\eql@totalrows@
658       +\ifdefined\eql@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\tagpos@best@use\eq@true
664     \eq\tagpos@init@out
665   \fi
666 }
```

TODO: describe

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

TODO: describe

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

TODO: describe

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

TODO: describe

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

TODO: describe

```
685 \let\eq\tagpos@init@centerone\eq\tagpos@init@continuous
686 \def\eq\tagpos@eval@centerone{%
687   \eq\tagpos@current@\dimexpr(\eq\tagpos@totalheight@
688     -\eq\tagpos@height@block@-\eq\tagpos@depth@block@)/\tw@\relax
689 }
```

TODO: describe

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

TODO: describe

```
695 \let\eq\tagpos@init@median\eq\tagpos@init@continuous
696 \def\eq\tagpos@eval@median{%
697   \ifnum\eq\tagpos@totalrows@=\z@
698     \ifodd\eq\tagpos@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\eq\@tags@set@label{#1}}%
824 }

```

TODO: describe

```

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

```

Tags. **TODO:** describe

\eq\@tag@default

```

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

```

\eq\@tag@gobble

```

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

```

TODO: describe

```

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

```

\eq\@tags@add@tagref

```

841 \def\eq\@tags@add@tagref[#1]#2{%
842 \def\eq\@tmp{#1}%
843 \ifx\eq\@tmp\eq\@testopt@default\else
844 \eq\@tags@add@ref{#1}%
845 \fi
846 \eq\@tags@add@tag{#2}%
847 }

```

TODO: describe

```

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

```

TODO: describe

```

856 \def\eq\@tags@set@ref#1{%
857 \ifdefined\eq\@tags@warn
858 \ifdefined\eq\@tags@ref
859 \eq\@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\eql@tags@addblock@tagform@off{%
887   \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
888 }

```

Raise Tags.

\raisetag

```

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

893 \def\eql@raisetag@gobble{%
894   \eql@amprotecttwo\eql@ifstar@tight\@gobble\@gobble
895 }

```

TODO: describe

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

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

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

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

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

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

E.4 Integration

TODO: describe

Support. **TODO:** describe

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

TODO: not necessary anymore

```
933 \def\eql@numbering@settools@gobble{%
934   \let\label\eql@label@gobble
935   \let\tag\eql@tag@gobble
936   \let\raisetag\eql@raisetag@gobble
937   \let\numberhere\relax
```

```

938 \let\numbernext\relax
939 }

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

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

```

Single Line. TODO: describe

```

970 \def\eql@numbering@single@init{%
971   \let\eql@numbering@multi\eql@false
972   \eql@numbering@settools
973   \eql@numbering@eqnswinit
974   \eql@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     \ifeqnsw
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   \ifeqnsw
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@\@ne
```

```

1078 \ifdim\ifdim\eq\tagpos@target@<\z@-\fi
1079 \eq\tagpos@target@<\glueexpr\eq\tagpos@snap\relax
1080 \eq\tagpos@target@\z@
1081 \fi
1082 }

```

TODO: describe

```

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

```

TODO: describe

```

1103 \def\eq\tagpos@adjust@eval@convert{%
1104 \eq\tagpos@current@\z@
1105 \eq\@dimensions@for{%
1106 \ifnum\eq\@row@<\eq\tagpos@row@
1107 \advance\eq\tagpos@current@\dimexpr\eq\@line@interline@
1108 +\eq\@line@height@+\eq\@line@depth@\relax
1109 \fi
1110 \ifnum\eq\@row@=\eq\tagpos@row@
1111 \advance\eq\tagpos@current@\dimexpr\eq\@line@interline@
1112 +\eq\@line@height@-\eq\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      \eql@tagpos@target@z@
1131    \fi
1132    \ifdim\eql@tagpos@target@>%
1133      \ifdefined\eql@tagsleft-1sp\relax\else\z@fi
1134      \eql@tagpos@row@\eql@row@
1135      \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1136    \fi
1137    \advance\dimen@-\dimexpr\eql@line@interline@
1138      +\eql@line@depth@+\eql@line@height@\relax
1139    \fi
1140    \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1141      \eql@tagpos@footroom@\eql@line@interline@
1142    \fi
1143  }%
1144 }

```

TODO: describe

```

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

```

TODO: describe

```

1152 \def\eql@tagpos@print@line@eval@continuous{%
1153   \if@eqnsw
1154     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1155     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1156     \eql@tagpos@plain@\eql@tagpos@target@
1157     \@tempdima\dimexpr\eql@line@height@+\eql@tagpos@headroom@
1158       -\ht\eql@tagbox@\relax
1159     \@tempdimb\dimexpr-\eql@line@depth@-\eql@tagpos@footroom@
1160       +\dp\eql@tagbox@\relax
1161     \ifnum\eql@row@=\@ne
1162       \@tempdima.5\maxdimen
1163     \fi
1164     \ifnum\eql@row@=\eql@totalrows@
1165       \@tempdimb-.5\maxdimen
1166     \fi
1167     \ifdim\eql@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\eq\tagpos@prevrow@>\z@
1185 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1186 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1187 \eq\tagpos@raised@\eq\tagpos@plain@
1188 \let\eq\tagpos@reserve\eq\false
1189 \fi
1190 \else
1191 \ifdim\eq\tagpos@target@>%
1192 \ifdefined\eq\tagsleft-1sp\relax\else\z@\fi
1193 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1194 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1195 \eq\tagpos@raised@\eq\tagpos@plain@
1196 \let\eq\tagpos@reserve\eq\false
1197 \fi
1198 \else
1199 \eq\tagpos@raised@\dimexpr-\eq\line@depth@
1200 -\ht\eq\tagbox@\relax
1201 \ifdim\eq\tagpos@raised@<\eq\tagpos@plain@\else
1202 \eq\tagpos@raised@\eq\tagpos@plain@
1203 \let\eq\tagpos@reserve\eq\false
1204 \fi
1205 \fi
1206 \fi
1207 \else
1208 \ifnum\eq\tagpos@prevrow@=\eq\row@
1209 \eq\tagwidth@\eq\tagwidth@block@
1210 \else
1211 \let\eq\tagpos@reserve\eq\false
1212 \fi
1213 \fi
1214 }

```

TODO: describe

```

1215 \def\eq\tagpos@print@line@eval@discrete{%
1216 \if@eqnsw
1217 \ht\eq\tagbox@\dimexpr\ht\eq\tagbox@-\eq\tagpos@smashup@\relax
1218 \dp\eq\tagbox@\dimexpr\dp\eq\tagbox@-\eq\tagpos@smashdown@\relax
1219 \eq\tagpos@plain@\eq\tagpos@shift@
1220 \eq\tagpos@headroom@\z@
1221 \eq\tagpos@footroom@\z@
1222 \ifdim\eq\tagpos@shift@>%
1223 \ifdefined\eq\tagsleft-1sp\relax\else\z@\fi
1224 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1225 \else
1226 \eq\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         +\eql@line@depth@\relax\relax}%
1239     \eql@displaybreak@star\@M
1240     \fi
1241     \fi
1242     \fi
1243 }

```

E.6 Component Display

Showkeys Integration. **TODO:** describe

```

1244 \let\eql@SK@loaded\eql@false
1245 \let\eql@SK@label\@gobble
1246 \let\eql@SK@clearlabel\@empty
1247 \let\eql@SK@setlabel\@gobble
1248 \let\eql@SK@printlabel@right\@empty
1249 \let\eql@SK@printlabel@left\@empty
1250 \let\eql@SK@printlabel@line\@empty
1251 \def\eql@label@clean{\eql@label@org}
1252 \AddToHook{package/showkeys/after}{
1253     \let\eql@SK@loaded\eql@true
1254     \def\eql@SK@label#1{\SK@\SK@@label#1}
1255     \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1256     \eql@SK@clearlabel
1257     \def\eql@SK@@label#1>#2\SK@{%
1258         \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1259     }
1260     \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1261     \def\eql@SK@printlabel@right{%
1262         \ifx\eql@SK@lab\relax\else
1263             \rlap{\kern\marginparsep\eql@SK@lab}%
1264             \eql@SK@clearlabel
1265         \fi
1266     }
1267     \def\eql@SK@printlabel@left{%
1268         \ifx\eql@SK@lab\relax\else
1269             \llap{\eql@SK@lab\kern\marginparsep}%
1270             \eql@SK@clearlabel
1271         \fi
1272     }
1273     \def\eql@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{\eq\@tags@glabel}%
1337 }%
1338 }

```

TODO: describe

`\eq\@composetag@anchor`

```

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

```

Tag Layout. **TODO:** describe

```

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

```

TODO: describe

```

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

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

1371 \protected\def\tagform{\eq\@tags@tagform}
1372 \protected\def\tagbox{\eq\@tags@taglayout}
1373 \protected\def\tagboxed{\eq\@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\eq\@eqref#1{\textup{\eq\@tags@tagcompose{\ref{#1}}}}

```

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

```

TODO: describe

```

1415 \def\eql@tagbox@print@adjustheadroom{%
1416   \dimen@ \dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1417   \ifdim\dimen@>\z@
1418     \ifdim\dimen@>\eql@tagpos@headroom@

```

```

1419      \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1420      \else
1421      \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1422      \fi
1423      \fi
1424 }

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

1449 \def\eql@tagbox@print@prepare{%
1450   \ifdefined\eql@tagpos@reserve
1451     \eql@tagpos@current@\eql@tagpos@plain@
1452   \else
1453     \eql@tagpos@current@\eql@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\eq\tagbox@print@tagsright{%
1469   \eq\tagbox@print@prepare
1470   \kern-\wd\eq\tagbox@
1471   \raise\eq\tagpos@current@\box\eq\tagbox@
1472 }
```

TODO: describe

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

eq\tagbox@print@cell

```
1479 \def\eq\tagbox@print@cell{%
1480   \eq\tagging@tagaddbox
1481   \ifdefined\eq\tagsleft
1482     \ifdefined\eq\tagpos@reserve
1483       \ifdim\eq\tagwidth@>\dimexpr\eq\line@avail@+\eq\tagfuzz@\relax
1484         \let\eq\tagpos@reserve\eq>false
1485       \fi
1486     \fi
1487     \if@eqnsw
1488       \eq\tagbox@print@tagsleft
1489     \fi
1490     \kern\displaywidth
1491   \else
1492     \kern\displaywidth
1493     \ifdefined\eq\tagpos@reserve
1494       \ifdim\eq\tagwidth@>%
1495         \dimexpr\displaywidth-\eq\line@width@+\eq\tagfuzz@\relax
1496         \let\eq\tagpos@reserve\eq>false
1497       \fi
1498     \fi
1499     \if@eqnsw
1500       \eq\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 \eq\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 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\eql@subequations@close{%
1532   \ifnum\c@equation=\z@
1533     \eql@subequations@restorecounter
1534   \else
1535     \global\c@equation\c@parentequation
1536   \fi
1537 }

```

F.2 Environment

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

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

```

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 }
```

`\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\eq@label@clean\expandafter{\eq@tags@label}%
1577       \endgroup
1578     \fi
1579   \fi
1580   \eq@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\eql@numbering@subeq@close{%
1623   \eql@subequations@close
1624 }

```

G Display Equations Support

TODO: describe

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

G.1 Display Breaks

TODO: describe

erdisplaylinepenalty

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

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

```
1633 \def\eql@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\eqldisplaybreak@default{%
1637   \eql@warning{Invalid use of \string\displaybreak{}}%
1638   \eql@teststaroropt@loose\@gobble\eql@gobbleopt{}}
1639 \eql@amsmath@after{\let\eqldisplaybreak@default\displaybreak}
1640 \eql@amsmath@let\displaybreak\eqldisplaybreak@default
```

```
1641 \newcount\eqldisplaybreak@pen@
1642 \newcount\eqldisplaybreak@prepen@
1643 \newcount\eqldisplaybreak@postpen@
```

TODO: describe

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

```
1649 \def\eqldisplaybreak@star#1{%
1650   \global\eql@appendexpand\eql@interline@container{%
1651     \eqldisplaybreak@pen@\the\numexpr#1\relax\relax}%
1652 }
```

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

TODO: describe

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

TODO: describe

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

TODO: describe

```
1675 \def\eqldisplaybreak@inter#1{%
1676   \ifnum#1<\z@
1677     \interdisplaylinepenalty\@M
1678   \else
1679     \interdisplaylinepenalty\eqldgetdsp@pen{#1}\relax
1680   \fi
1681 }
```

G.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```
1682 \newskip\eqlvspaceskip@
1683 \let\eqlvspace@org\vspace
1684 \def\eqlvspace{%
1685   \ifvmode
1686     \expandafter\eqlvspace@immediate
1687   \else
1688     \expandafter\eqlvspace@line
1689   \fi
1690 }
```

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

```
1691 \def\eqlvspace@line{%
1692   \eq@ifstar@loose\eqlvspace@addfixedbefore\eqlvspace@add
1693 }
1694 \def\eqlvspace@add#1{%
1695   \global\eql@appendexpand\eql@interline@container{%
1696     \advance\eqlvspaceskip@\the\glueexpr#1\relax\relax}}
1697 \def\eqlvspace@addfixedbefore#1{%
1698   \global\eql@appendexpand\eql@interline@container{%
1699     \noexpand\eql@append\noexpand\eqldisplay@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

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

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

```
1781 \newskip\eql@abovespace@
1782 \newskip\eql@belowspace@
```

`\eql@display@enter`

```
1783 \def\eql@display@enter{%
1784   \if@noskipsec\leavevmode\par\fi
1785   \ifvmode
1786     \eql@prevdepth@\prevdepth
1787     \nointerlineskip
1788     \noindent
1789   \else
1790     \eql@prevdepth@\maxdimen
1791   \fi
1792 }
```

`\eql@display@adjust`

```

1793 \def\eql@display@adjust{%
1794   \ifdefined\eql@display@linewidth
1795     \displaywidth\glueexpr\eql@display@linewidth\relax
1796     \advance\displaywidth-\displayindent
1797   \fi
1798   \ifdefined\eql@display@marginleft
1799     \advance\displaywidth\displayindent
1800     \displayindent\glueexpr\eql@display@marginleft\relax
1801     \advance\displaywidth-\displayindent
1802   \fi
1803   \ifdefined\eql@display@marginright
1804     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1805   \fi
1806   \ifdim\displaywidth<\z@
1807     \displaywidth\z@
1808   \fi
1809 }

```

`\eql@display@init`

```

1810 \def\eql@display@init{%
1811   \let\displaybreak\eql@displaybreak
1812   \let\eql@vspace@org\vspace
1813   \let\vspace\eql@vspace
1814   \let\eqncontrol\eql@control
1815   \let\eql@display@injectbefore\@empty
1816   \let\eql@display@injectafter\@empty
1817   \eql@spread@set
1818   \eql@strut@make
1819   \let\eql@frame@cmd\@undefined
1820 }

```

`\eql@display@print`

```

1821 \def\eql@display@print{%
1822   \let\eql@display@container\@empty
1823   \eql@display@firstavail\z@
1824   \eql@display@aboveextend\z@
1825   \eql@display@belowextend\z@
1826   \global\let\eql@interline@container\eql@interline@container@clear
1827 }

```

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

```

1828 \def\eql@display@halign@init#1{%
1829   \eql@row\z@
1830   \eql@prevgraf\prevgraf
1831   \everycr{\noalign{%
1832     \global\advance\eql@row@\@ne
1833     \prevgraf\numexpr\prevgraf+\@ne\relax
1834     #1%
1835   }}%
1836 }

```

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

```

1837 \def\eql@display@halign@start{%
1838   \prevgraf\numexpr\eql@prevgraf+\@ne\relax

```

```

1839 \ifdim\eq\@prevdepth@=\maxdimen\else
1840   \prevdepth\eq\@prevdepth@
1841 \fi
1842 \ifdim\prevdepth=-\@m\p@\else
1843   \ifdefined\eq\@display@height
1844     \skip@\baselineskip
1845     \advance\skip@-\glueexpr\eq\@display@height\relax
1846     \advance\skip@-\prevdepth\relax
1847     \ifdim\skip@<\lineskiplimit
1848       \skip@\lineskip
1849     \fi
1850     \advance\skip@-\eq\@spread@\relax
1851     \vskip\skip@
1852     \nointerlineskip
1853   \else
1854     \vskip-\eq\@spread@\relax
1855   \fi
1856 \fi
1857 }

```

TODO: describe

```

1858 \def\eq\@display@vspace{%
1859   \advance\abovedisplayskip\eq\@abovespace@
1860   \advance\belowdisplayskip\eq\@belowspace@
1861 }

```

TODO: describe

```

1862 \def\eq\@display@vspace@native{%
1863   \advance\abovedisplayskip\eq\@abovespace@
1864   \advance\belowdisplayskip\eq\@belowspace@
1865   \advance\abovedisplayshortskip\eq\@abovespace@
1866   \advance\belowdisplayshortskip\eq\@belowspace@
1867 }

```

TODO: describe

```

1868 \def\eq\@display@penalty{%
1869   \ifnum\eq\@displaybreak@postpen@=\@MM\else
1870     \postdisplaypenalty\eq\@displaybreak@postpen@
1871   \fi
1872   \ifnum\eq\@displaybreak@open@=\@MM\else
1873     \postdisplaypenalty\eq\@displaybreak@open@
1874   \fi
1875   \ifnum\eq\@displaybreak@prepen@=\@MM\else
1876     \predisplaypenalty\eq\@displaybreak@prepen@
1877   \fi
1878 }

```

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

```

1879 \def\eq\@display@halign@end{%
1880   \eq\@interline@container
1881   \eq\@display@injectbefore
1882   \global\eq\@prevgraf@\numexpr\prevgraf+\@ne\relax
1883   \ifdefined\eq\@display@depth
1884     \prevdepth\glueexpr\eq\@display@depth\relax
1885   \fi
1886 }

```

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

```

1887 \def\eql@display@close{%
1888   \eql@display@container
1889   \ifdim\eql@display@firstavail@<\z@
1890     \eql@display@firstavail@\z@
1891   \fi
1892   \eql@skip@mode@leave@\z@
1893   \ifdim\eql@prevdepth@=\maxdimen
1894     \ifdim\predisplaysize=-\maxdimen
1895       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1896       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1897     \else
1898       \eql@skip@mode@above@\z@
1899       \eql@skip@mode@below@\z@
1900       \advance\eql@display@firstavail@\displayindent
1901       \ifdim\eql@display@firstavail@>\predisplaysize
1902         \ifcase\eql@skip@mode@short\relax
1903         \or
1904           \eql@skip@mode@above@\@ne
1905         \or
1906           \eql@skip@mode@above@\@ne
1907           \ifnum\eql@totalrows@=\@ne
1908             \eql@skip@mode@below@\@ne
1909           \fi
1910         \or
1911           \eql@skip@mode@above@\@ne
1912           \eql@skip@mode@below@\@ne
1913         \fi
1914       \fi
1915     \fi
1916   \else
1917     \ifdim\eql@prevdepth@=-\@m\p@
1918       \eql@skip@mode@above@\eql@skip@mode@top@above\relax
1919       \eql@skip@mode@below@\eql@skip@mode@top@below\relax
1920     \else
1921       \eql@skip@mode@above@\eql@skip@mode@par@above\relax
1922       \eql@skip@mode@below@\eql@skip@mode@par@below\relax
1923     \fi
1924   \fi
1925   \ifcase\eql@skip@mode@above@
1926   \or\or\or
1927     \predisplaypenalty\z@
1928   \or
1929     \predisplaypenalty\z@
1930   \fi
1931   \ifcase\eql@skip@mode@below@
1932   \or\or\or
1933     \eql@skip@mode@leave@\@ne
1934   \or
1935     \eql@skip@mode@leave@\tw@
1936   \fi
1937   \ifdefined\eql@skip@force@above
1938     \eql@skip@mode@above@\eql@skip@force@above\relax
1939   \fi
1940   \ifdefined\eql@skip@force@below
1941     \eql@skip@mode@below@\eql@skip@force@below\relax

```

```

1942 \fi
1943 \ifdefined\eq@skip@force@leave
1944   \eq@skip@mode@leave@\eq@skip@force@leave\relax
1945 \fi
1946 \ifnum\eq@skip@mode@leave@>\z@
1947   \postdisplaypenalty\z@
1948 \fi
1949 \ifcase\eq@skip@mode@above@
1950   \abovedisplayskip\glueexpr\eq@skip@long@above\relax
1951 \or
1952   \abovedisplayskip\glueexpr\eq@skip@short@above\relax
1953 \or
1954   \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
1955 \or
1956   \abovedisplayskip\glueexpr\eq@skip@par@above\relax
1957 \or
1958   \abovedisplayskip\glueexpr\eq@skip@top@above\relax
1959 \or
1960   \abovedisplayskip\z@skip
1961 \or
1962   \abovedisplayskip\glueexpr\eq@skip@med@above\relax
1963 \or
1964   \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
1965 \fi
1966 \ifcase\eq@skip@mode@below@
1967   \belowdisplayskip\glueexpr\eq@skip@long@below\relax
1968 \or
1969   \belowdisplayskip\glueexpr\eq@skip@short@below\relax
1970 \or
1971   \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
1972 \or
1973   \belowdisplayskip\glueexpr\eq@skip@par@below\relax
1974 \or
1975   \belowdisplayskip\glueexpr\eq@skip@top@below\relax
1976 \or
1977   \belowdisplayskip\z@skip
1978 \or
1979   \belowdisplayskip\glueexpr\eq@skip@med@below\relax
1980 \or
1981   \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
1982 \fi
1983 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
1984 \eq@interline@container
1985 \advance\eq@belowspace@\eq@vspaceskip@
1986 \eq@display@penalty
1987 \eq@display@vspace
1988 \skip@\glueexpr\eq@skip@tag@above\relax
1989 \ifdim\skip@>\abovedisplayskip
1990   \skip@\abovedisplayskip
1991 \fi
1992 \advance\abovedisplayskip-\eq@display@aboveextend@\relax
1993 \ifdim\abovedisplayskip<\skip@
1994   \abovedisplayskip\skip@
1995 \fi
1996 \skip@\glueexpr\eq@skip@tag@below\relax
1997 \ifdim\skip@>\belowdisplayskip
1998   \skip@\belowdisplayskip
1999 \fi

```

```

2000 \ifdim\eqldisplay@belowextend@>\z@
2001 \advance\belowdisplayskip-\eqldisplay@belowextend@relax
2002 \ifdim\belowdisplayskip<\skip@
2003 \belowdisplayskip\skip@
2004 \fi
2005 \fi
2006 }

```

TODO: describe

```

2007 \def\eqldisplay@leave{%
2008 \prevgraf\eql@prevgraf@
2009 \ifcase\eql@skip@mode@leave@
2010 \or
2011 \endgraf
2012 \or
2013 \endgraf
2014 \prevdepth-\@m\p@
2015 \fi
2016 }

```

TODO: describe

```

2017 \def\eqldisplay@nest{%
2018 \let\displaybreak\eqldisplaybreak@default
2019 \let\intertext\eql@intertext@default
2020 \let\vspace\eql@vspace@org
2021 }

```

TODO: describe

```

2022 \def\eqldisplay@restore{%
2023 \let\label\eql@label@org
2024 \let\tag\eql@tag@default
2025 \let\raisetag\eql@raisetag@default
2026 \let\displaybreak\eqldisplaybreak@default
2027 \let\intertext\eql@intertext@default
2028 \let\vspace\eql@vspace@org
2029 }

```

TODO: describe

```

2030 \eql@append\arrayparboxrestore{%
2031 \eqldisplay@restore
2032 \ifdefined\eql@ampproof@active
2033 \eql@amprevert
2034 \fi
2035 \@displayfalse
2036 }

```

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)

```

2037 \def\eql@stack@enable{%
2038 \let\eql@stack@save@equations\eql@stack@save@equations@
2039 \let\eql@stack@save@box\eql@stack@save@box@
2040 }

```

TODO: describe

```
2041 \let\eq\stack@save@equations\eq\stack@enable
2042 \let\eq\stack@save@box\eq\stack@enable
2043 \let\eq\stack@restore\@empty
```

TODO: describe

```
2044 \def\eq\stack@save@reg#1{\global#1\the#1\relax}
2045 \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

```
2046 \def\eq\stack@save@equations{%
2047   \let\eq\stack@numbering@eqnswinit\eq\numbering@eqnswinit
2048   \let\eq\stack@cell@container\eq\cell@container
2049   \let\eq\stack@tags@container\eq\tags@container
2050   \let\eq\stack@interline@container\eq\interline@container
2051   \let\eq\stack@block@container\eq\display@container
2052   \let\eq\stack@dimensions@tab\eq\dimensions@tab
2053   \edef\eq\stack@restore{%
2054     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2055     \eq\stack@save@let\eq\stack@numbering@eqnswinit\eq\numbering@eqnswinit
2056     \eq\stack@save@let\eq\stack@cell@container\eq\cell@container
2057     \eq\stack@save@let\eq\stack@tags@container\eq\tags@container
2058     \eq\stack@save@let\eq\stack@interline@container\eq\interline@container
2059     \eq\stack@save@let\eq\stack@dimensions@tab\eq\dimensions@tab
2060     \eq\stack@save@let\eq\stack@block@container\eq\display@container
2061     \eq\stack@save@reg\eq\column@
2062     \eq\stack@save@reg\eq\totalcolumns@
2063     \eq\stack@save@reg\eq\line@avail@
2064     \eq\stack@save@reg\eq\line@pos@
2065     \eq\stack@save@reg\eq\line@width@
2066     \eq\stack@save@reg\eq\line@depth@
2067     \eq\stack@save@reg\eq\line@height@
2068     \eq\stack@save@reg\eq\line@prevdepth@
2069     \eq\stack@save@reg\eq\line@interline@
2070     \eq\stack@save@reg\eq\totalheight@
2071     \eq\stack@save@reg\eq>tagwidth@max@
2072     \eq\stack@save@reg\eq>tagpos@row@
2073     \eq\stack@save@reg\eq)row@
2074     \eq\stack@save@reg\eq>tagrows@
2075   }%
2076 }
```

TODO: describe

```
2077 \def\eq\stack@save@box{%
2078   \let\eq\stack@cell@container\eq\cell@container
2079   \edef\eq\stack@restore{%
2080     \eq\stack@save@let\eq\stack@cell@container\eq\cell@container
2081     \eq\stack@save@reg\eq)row@
2082   }%
2083 }
```


H Multi-Line Support

TODO: describe

H.1 Measure Support

TODO: describe

```
2084 \def\eq@measure@init#1#2{%
2085   \eq@dimensions@reset
2086   \let\eq@display@container\@empty
2087   \eq@numbering@measure@init
2088   \eq@row@\z@
2089   \eq@totalheight@\z@
2090   \eq@totalrows@\@M
2091   \eq@line@prevdepth@-\@m\p@
2092   \eq@line@interline@\z@
2093   \tabskip\z@skip
2094   \everycr{\noalign{%
2095     \global\advance\eq@row@\@ne
2096     #1%
2097   }}%
2098   \global\let\eq@interline@container\eq@interline@container@clear
2099   \eq@measure@savestate
2100   \eq@display@halign@letcr{#2}%
2101 }
```

TODO: describe

```
2102 \def\eq@measure@tag{%
2103   \eq@tagwidth@\z@
2104   \ifdefined\eq@numbering@multi
2105     \if@eqnsw
2106       \eq@tags@container
2107       \eq@tagbox@make\eq@composetag@measure
2108       \ifdefined\eq@tagpos@reserve\else
2109         \eq@tagwidth@\z@
2110       \fi
2111     \fi
2112   \fi
2113 }
```

TODO: describe

```
2114 \def\eq@measure@endrow{%
2115   \ifdim\eq@line@prevdepth@=-\@m\p@\else
2116     \dimen@\dimexpr\baselineskip-\eq@line@height@-\eq@line@prevdepth@\relax
2117     \ifdim\dimen@<\lineskiplimit
2118       \dimen@\lineskip
2119     \fi
2120     \advance\eq@line@interline@\dimen@
2121   \fi
2122   \eq@dimensions@endrow
2123   \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2124     \global\eq@tagwidth@max@\eq@tagwidth@
2125   \fi
2126   \ifdim\eq@tagwidth@>\z@
2127     \global\advance\eq@tagrows@\@ne
```

```

2128 \fi
2129 \global\advance\eql@totalheight@\dimexpr
2130 \eql@line@interline@+\eql@line@height@+\eql@line@depth@
2131 \global\eql@line@interline@\z@
2132 \global\eql@line@prevdepth@\eql@line@depth@
2133 }

```

TODO: describe

```

2134 \def\eql@measure@close{%
2135 \advance\eql@row@-\tw@
2136 \eql@totalrows@\eql@row@
2137 \ifnum\eql@totalrows@>\z@
2138 \eql@dimensions@get@one
2139 \eql@topheight@\dimexpr\eql@line@height@+\eql@line@interline@\relax
2140 \eql@dimensions@get\eql@totalrows@
2141 \eql@bottomdepth@\eql@line@depth@
2142 \fi
2143 \eql@numbering@measure@blocktag
2144 \begingroup
2145 \eql@tags@container
2146 \if@eqnsw
2147 \eql@tagbox@make\eql@composetag@measure
2148 \ifdefined\eql@tagpos@reserve\else
2149 \eql@tagwidth@\z@
2150 \fi
2151 \eql@dimensions@saveblocktag
2152 \else
2153 \eql@dimensions@savenoblocktag
2154 \eql@numbering@warnunused
2155 \fi
2156 \endgroup
2157 \eql@dimensions@get\z@
2158 \eql@measure@restorestate
2159 }

```

measure@restorestate

eql@measure@savestate

```

2160 \let\eql@measure@restorestate\@empty
2161 \def\eql@measure@savestate{%
2162 \begingroup
2163 \def\@elt##1{%
2164 \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2165 \global\edef\@gtempa{\cl@@ckpt}%
2166 \endgroup
2167 \let\eql@measure@restorestate\@gtempa
2168 }

```

H.2 Line Breaks

TODO: describe

\eql@display@cr

```

2169 \protected\def\eql@display@cr{%
2170 \eql@ampprotecttwo\eql@teststaropt@tight{%
2171 \global\eql@append\eql@interline@container{\eql@displaybreak@pen@\@M}%
2172 \eql@display@cr@opt}

```

```

2173 \eqldisplay@cr@opt\z@skip
2174 }

```

`\eqldisplay@cr@opt`

```

2175 \def\eqldisplay@cr@opt[#1]{%
2176 \eqldisplay@endline
2177 \cr

2178 \noalign{%
2179 \eql@interline@container
2180 \eqldisplay@injectbefore
2181 \ifnum\eqldisplaybreak@pen@=\@MM
2182 \penalty\interdisplaylinepenalty
2183 \else
2184 \penalty\eqldisplaybreak@pen@
2185 \fi
2186 \advance\eql@vspaceskip@\glueexpr#1\relax
2187 \vskip\eql@vspaceskip@
2188 \global\advance\eql@line@interline@\eql@vspaceskip@
2189 \eqldisplay@injectafter
2190 \global\let\eql@interline@container\eql@interline@container@clear
2191 }%
2192 }

```

`display@halign@letcr`

```

2193 \def\eqldisplay@halign@letcr#1{%
2194 \let\\\eqldisplay@cr
2195 \let\eqldisplay@endline#1%
2196 }

```

H.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2197 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2198 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2199 \def\eql@intertext@process{%
2200 \eqldisplay@endline
2201 \cr
2202 \ifmeasuring@
2203 \expandafter\@gobble
2204 \else
2205 \expandafter\eql@intertext@print
2206 \fi
2207 }

```

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

```

2208 \def\eql@intertext@print#1{%
2209 \noalign{%
2210 \eqldisplay@halign@end
2211 \let\eql@skip@force@below\z@

```

```

2212 \let\eq\@skip@force@above\z@
2213 \eq\@setkeys{intertext}\eq\intertext@opt
2214 \openup-\eq\spread@
2215 \penalty\postdisplaypenalty
2216 \ifcase\eq\@skip@force@below\relax
2217   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@long@below\relax
2218 \or
2219   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@short@below\relax
2220 \or
2221   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@cont@below\relax
2222 \or
2223   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@par@below\relax
2224 \or
2225   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@top@below\relax
2226 \or
2227   \advance\eq\@vspaceskip@\z@skip
2228 \or
2229   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@med@below\relax
2230 \or
2231   \advance\eq\@vspaceskip@\glueexpr\eq\@skip@custom@below\relax
2232 \fi
2233 \vskip\eq\@vspaceskip@
2234 \global\let\eq\@interline@container\eq\@interline@container@clear
2235 \vbox{%
2236   \@parboxrestore
2237   \ifdim
2238     \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2239   \else
2240     \parshape\@ne
2241     \@totalleftmargin\linewidth
2242   \fi
2243   \noindent
2244   \prevgraf\eq\@prevgraf@
2245   \ignorespaces
2246   #1%
2247   \par
2248   \global\eq\@prevgraf@\prevgraf
2249 }%
2250 \penalty\predisplaypenalty
2251 \ifcase\eq\@skip@force@above\relax
2252   \vskip\glueexpr\eq\@skip@long@above\relax
2253 \or
2254   \vskip\glueexpr\eq\@skip@short@above\relax
2255 \or
2256   \vskip\glueexpr\eq\@skip@cont@above\relax
2257 \or
2258   \vskip\glueexpr\eq\@skip@par@above\relax
2259 \or
2260   \vskip\glueexpr\eq\@skip@top@above\relax
2261 \or
2262   \vskip\z@skip
2263 \or
2264   \vskip\glueexpr\eq\@skip@med@above\relax
2265 \or
2266   \vskip\glueexpr\eq\@skip@custom@above\relax
2267 \fi
2268 % \eq\@prevdepth@\maxdimen
2269 \eq\@prevdepth@\z@

```

```

2270 \eqldisplay@halign@start
2271 }
2272 }

```

TODO: describe

```

2273 \newenvironment{eqldintertext}{%
2274 \eqldtestopt@tight\eqldintertext@{}%
2275 }{%
2276 \aftergroup\eqldintertext@after
2277 \ignorespacesafterend
2278 }

```

TODO: describe

```

2279 \def\eqldintertext@env{intertext}
2280 \def\eqldintertext@[#1]{%
2281 \global\def\eqldintertext@opt{#1}%
2282 \ifx\@currenvir\eqldintertext@env
2283 \expandafter\eqldscan@env\expandafter\eqldintertext@inject
2284 \else
2285 \expandafter\eqldintertext@process
2286 \fi
2287 }

```

TODO: describe

```

2288 \def\eqldintertext@inject{%
2289 \global\edef\eqldintertext@after{%
2290 \noexpand\eqldintertext@process{%
2291 \ifx\eqldscan@body\eqldscan@body@dump
2292 \eqldscan@body@dump
2293 \else
2294 \noexpand\scantokens{\eqldscan@body@dump}%
2295 \fi
2296 }%
2297 }%
2298 }

```

H.4 Line Marks

TODO: describe

```

2299 \def\eqldmarkline@pos@below{below}
2300 \def\eqldmarkline@pos@bottom{bottom}
2301 \def\eqldmarkline@pos@baseline{baseline}
2302 \let\eqldmarkline@pos\eqldmarkline@pos@baseline
2303 \let\eqldmarkline@shift\z@
2304 \def\eqldmarkline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2305 \def\eqldmarkline@symbol{}

```

TODO: describe

```

2306 \def\eqldmarkline@select#1{%
2307 \let\eqldmarkline@shift\z@
2308 \eqldsetkeys{markline}{#1}%
2309 \eqldmarkline@print
2310 }

```

TODO: describe

```

2311 \def\eq@markline@print{%
2312   \dimen@ \dimexpr\eq@markline@shift\relax
2313   \ifx\eq@markline@pos\eq@markline@pos@below
2314     \ifdim\dimen@=\z@\else
2315       \penalty\@M
2316       \vskip-\dimen@
2317     \fi
2318     \nointerlineskip
2319     \penalty\@M
2320     \vbox{\hfill\hbox{\eq@markline@symbol}}}%
2321   \else
2322     \ifx\eq@markline@pos\eq@markline@pos@baseline
2323       \advance\dimen@\prevdepth
2324     \fi
2325     \setbox\z@\hbox{\raise\dimen@\hbox{\eq@markline@symbol}}}%
2326     \dimen@\prevdepth
2327     \ht\z@\z@
2328     \dp\z@\z@
2329     \nointerlineskip
2330     \penalty\@M
2331     \vbox{\hfill\box\z@}%
2332     \prevdepth\dimen@
2333   \fi
2334 }

```

TODO: describe

```

2335 \def\eq@markline@inject#1{%
2336   \let\eq@markline@push\eq@false
2337   \ifx\eq@markline@pos\eq@markline@pos@below\else
2338     \ifdefined\eq@tagsleft\else
2339       \ifx\eq@equations@main\eq@multi@main
2340         \ifdefined\eq@numbering@multi
2341           \if@eqnsw
2342             \let\eq@markline@push\eq@true
2343           \fi
2344         \else
2345           \ifnum\eq@row@=\eq@tagpos@row@
2346             \let\eq@markline@push\eq@true
2347           \fi
2348         \fi
2349       \else
2350         \if@eqnsw
2351           \let\eq@markline@push\eq@true
2352         \fi
2353       \fi
2354     \fi
2355   \fi
2356   \ifdefined\eq@markline@push
2357     \global\eq@append\eq@interline@container{%
2358       \eq@append\eq@display@injectbefore{\eq@markline@select{push,#1}}}%
2359   \else
2360     \global\eq@append\eq@interline@container{%
2361       \eq@append\eq@display@injectbefore{\eq@markline@select{#1}}}%
2362   \fi
2363 }

```

TODO: describe

```

2364 \def\eq@markline@amsthm@opt[#1]{\eq@markline@inject{qed,#1}}

```

```

2365 \def\eq\@markline\@amsthm\@staropt[#1]{\eq\@markline\@inject{\qed,push,#1}}
2366 \def\eq\@markline\@amsthm\@qed{\eq\@teststaropt\@tight
2367   \eq\@markline\@amsthm\@staropt\eq\@markline\@amsthm\@opt{}}
2368 \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 $\text{eq\@shape\@amount@}$ specify the currently selected horizontal alignment (0 for left, 1 for center, 2 for right) and the indentation amount, respectively:

```

2369 \newcount\eq\@shape\@pos@
2370 \newdimen\eq\@shape\@amount@
2371 \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

$\text{eq\@marginleft\@min@}$ (*dimen*)
 eq\@marginright@ (*dimen*)
 eq\@centeroffset@ (*dimen*)

```

2372 \newdimen\eq\@marginleft@
2373 \newdimen\eq\@marginright@
2374 \newdimen\eq\@marginleft\@min@
2375 \newdimen\eq\@centeroffset@

```

I.2 Shape Schemes

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

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

```

2376 \def\eq\@shape\@tab\@default{\default}
2377 \def\eq\@shape\@tab\@left{left}
2378 \def\eq\@shape\@tab\@center{center}
2379 \def\eq\@shape\@tab\@right{right}
2380 \def\eq\@shape\@tab\@first{first}
2381 \def\eq\@shape\@tab\@hanging{hanging}
2382 \def\eq\@shape\@tab\@steps{steps}

```

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

```

2383 \let\eq\@shape\@tab\@def\eq\@shape\@tab\@default
2384 \let\eq\@shape\@tab\@eq\@shape\@tab\@default
2385 \let\eq\@shape\@tab\@l\eq\@shape\@tab\@left
2386 \let\eq\@shape\@tab\@c\eq\@shape\@tab\@center
2387 \let\eq\@shape\@tab\@r\eq\@shape\@tab\@right
2388 \let\eq\@shape\@tab\@rc\eq\@shape\@tab\@first
2389 \let\eq\@shape\@tab\@indent\eq\@shape\@tab\@first
2390 \let\eq\@shape\@tab\@hang\eq\@shape\@tab\@hanging
2391 \let\eq\@shape\@tab\@lc\eq\@shape\@tab\@hanging
2392 \let\eq\@shape\@tab\@outdent\eq\@shape\@tab\@hanging
2393 \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:

```
2394 \let\eq@shape@mode\eq@shape@tab@default
```

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

```
2395 \def\eq@shape@set#1{%
2396   \ifcsname eq@shape@tab@#1\endcsname
2397     \expandafter\let\expandafter\eq@shape@mode
2398     \csname eq@shape@tab@#1\endcsname
2399   \else
2400     \eq@error{shape '1' unknown: setting to default}%
2401     \let\eq@shape@mode\eq@shape@tab@default
2402   \fi
2403 }
```

`\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:

```
2404 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@ \eq@shape@amount@z@}
2405 \def\eq@shape@layoutcenter@center{\eq@shape@pos@one \eq@shape@amount@z@}
2406 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@ \eq@shape@amount@z@}
2407 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2408 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@ \eq@shape@amount@z@}
2409 \def\eq@shape@layoutleft@center{\eq@shape@pos@one \eq@shape@amount@z@}
2410 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@ \eq@shape@amount@z@}
2411 \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):

```
2412 \def\eq@shape@layoutcenter@first{%
2413   \eq@shape@pos@z@
2414   \eq@shape@amount@z@
2415   \ifnum\eq@totalrows@>\@ne
2416     \ifnum\eq@row@=\@ne
2417       \eq@shape@amount@\eq@indent@
2418     \fi
2419   \fi
2420 }
2421 \def\eq@shape@layoutleft@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 }
```

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:

```
2430 \def\eq@shape@layoutcenter@hanging{%
2431   \eq@shape@pos@z@
2432   \eq@shape@amount@\eq@indent@
2433   \ifnum\eq@totalrows@>\@ne
2434     \ifnum\eq@row@=\@ne
```



```

2435     \eql@shape@amount@\z@
2436   \fi
2437 \fi
2438 }
2439 \def\eql@shape@layoutleft@hanging{%
2440   \eql@shape@pos@\z@
2441   \eql@shape@amount@\z@
2442   \ifnum\eql@totalrows@>\@ne
2443     \ifnum\eql@row@=\@ne
2444       \eql@shape@amount@-\eql@indent@
2445     \fi
2446   \fi
2447 }

```

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:

```

2448 \def\eql@shape@layoutcenter@steps{%
2449   \eql@shape@amount@\z@
2450   \eql@shape@pos@\@ne
2451   \ifnum\eql@totalrows@>\@ne
2452     \ifnum\eql@row@=\@ne
2453       \eql@shape@pos@\z@
2454     \fi
2455     \ifnum\eql@row@=\eql@totalrows@
2456       \eql@shape@pos@\tw@
2457     \fi
2458   \fi
2459 }
2460 \def\eql@shape@layoutleft@steps{%
2461   \eql@shape@pos@\z@
2462   \eql@shape@amount@\z@
2463   \ifnum\eql@totalrows@>\@ne
2464     \ifnum\eql@row@=\@ne
2465       \eql@shape@amount@-\eql@indent@
2466     \fi
2467     \ifnum\eql@row@=\eql@totalrows@
2468       \eql@shape@amount@\eql@indent@
2469     \fi
2470   \fi
2471 }

```

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

```

2472 \let\eql@shape@eval\undefined
2473 \def\eql@shape@select{%
2474   \expandafter\let\expandafter\eql@shape@eval
2475   \csname eql@shape%
2476     @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
2477     @\eql@shape@mode\endcsname
2478 }

```

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

```

2479 \protected\def\eql@shape@alignleft{%
2480   \global\eql@append\eql@cell@container{\eql@shape@pos@\z@}%

```

```

2481 \eq\ampprotect\eq\shape@align@testpar\eq\shape@alignamount@opt}
2482 \protected\def\eq\shape@aligncenter{%
2483 \global\eq\append\eq\cell@container{\eq\shape@pos@\@ne}%
2484 \eq\ampprotect\eq\shape@align@testpar\eq\shape@alignamount@opt}
2485 \protected\def\eq\shape@alignright{%
2486 \global\eq\append\eq\cell@container{\eq\shape@pos@\@two}%
2487 \eq\ampprotect\eq\shape@align@testpar\eq\shape@alignamount@opt}
2488 \def\eq\shape@align@testpar#1{%
2489 \eq@ifstar@tight{#1[\eq\indent@]}%
2490 {\eq@ifnextgobble@tight{!}{#1[-\eq\indent@]}%
2491 {\eq@testopt@tight{#1}\z@}}
2492 \def\eq\shape@alignamount@opt[#1]{\eq\shape@alignamount@set{#1}}

```

$\text{eq}\text{shape@alignamount}$ **TODO:** describe

```

2493 \protected\def\eq\shape@alignamount{%
2494 \eq\ampprotecttwo\eq@ifstar@tight
2495 \eq\shape@alignamount@set\eq\shape@alignamount@add}
2496 \def\eq\shape@alignamount@add#1{%
2497 \global\eq\appendexpand\eq\cell@container{%
2498 \advance\eq\shape@amount@the\glueexpr#1\relax\relax}}
2499 \def\eq\shape@alignamount@set#1{%
2500 \global\eq\appendexpand\eq\cell@container{%
2501 \eq\shape@amount@the\glueexpr#1\relax\relax}}
2502 \def\eq\shape@align@enable{%
2503 \let\shoveleft\eq\shape@alignleft
2504 \let\shovecenter\eq\shape@aligncenter
2505 \let\shoveright\eq\shape@alignright
2506 \let\shoveby\eq\shape@alignamount
2507 }

```

TODO: describe

```

2508 \protected\def\eq\shape@align@default{%
2509 \eq\warn@here{\shove...}}%
2510 \eq\ampprotect\eq\shape@align@testpar\eq@gobbleopt}
2511 \protected\def\eq\shape@alignamount@default{%
2512 \eq\warn@here{\shove...}}%
2513 \eq\ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2514 \def\eq\shape@align@disable{%
2515 \let\shoveleft\eq\shape@align@default
2516 \let\shovecenter\eq\shape@align@default
2517 \let\shoveright\eq\shape@align@default
2518 \let\shoveby\eq\shape@alignamount@default
2519 }

```

I.3 Width Data

width@block@ (*dimen*)

```

2520 \newdimen\eq>tagwidth@block@
2521 \newdimen\eq>tagheight@block@
2522 \newdimen\eq>tagdepth@block@

```

$\text{eq}\text{dimensions@tab}$ **TODO:** new

```

2523 \let\eq\dimensions@tab\@empty

```

eql@dimensions@reset

```
2524 \def\eql@dimensions@reset{%
2525   \let\eql@dimensions@tab\@empty
2526   \eql@tagwidth@max@z@
2527   \eql@tagrows@z@
2528 }
```

\eql@dimensions@add

```
2529 \def\eql@dimensions@add#1{%
2530   \global\eql@appendexpand\eql@dimensions@tab{#1}%
2531 }
```

eql@dimensions@addreg

```
2532 \def\eql@dimensions@addreg#1{#1\the#1\relax}
```

@dimensions@startrow

```
2533 \def\eql@dimensions@startrow{%
2534   \eql@dimensions@add{\eql@dimensions@addreg\eql@row@}%
2535 }
```

@dimensions@savecell

```
2536 \def\eql@dimensions@savecell{%
2537   \eql@dimensions@add{%
2538     \eql@dimensions@addreg\eql@shape@pos@
2539     \eql@dimensions@addreg\eql@cellwidth@
2540     \eql@dimensions@addreg\eql@shape@amount@
2541     \noexpand\eql@dimensions@cellcall
2542   }%
2543 }
```

l@dimensions@savesep

```
2544 \def\eql@dimensions@savesep{%
2545   \eql@dimensions@add{\noexpand\eql@dimensions@sepcall}%
2546 }
```

eql@dimensions@endrow

```
2547 \def\eql@dimensions@endrow{%
2548   \eql@dimensions@add{,%
2549     \eql@dimensions@addreg\eql@tagwidth@
2550     \eql@dimensions@addreg\eql@line@height@
2551     \eql@dimensions@addreg\eql@line@depth@
2552     \eql@dimensions@addreg\eql@line@interline@
2553   };}%
2554 }
```

ensions@saveblocktag

```
2555 \def\eql@dimensions@saveblocktag{%
2556   \eql@dimensions@add{\eql@row@0\relax,%
2557     \eql@tagwidth@block@\the\eql@tagwidth@\relax
2558     \eql@tagheight@block@\the\ht\eql@tagbox@\relax
2559     \eql@tagdepth@block@\the\dp\eql@tagbox@\relax
```

```

2560 \eqldimensions@addreg\eqldtagpos@shift@
2561 \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
2562 \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
2563 ;}%
2564 \global\eqldtagwidth@max@\eqldtagwidth@
2565 \global\eqldtagrows@\@ne
2566 }

```

sions@savenoblocktag

```

2567 \def\eqldimensions@savenoblocktag{%
2568 \eqldimensions@add{\eqldrow@0\relax,;}%
2569 }

```

\eqldimensions@for

```

2570 \def\eqldimensions@for#1{%
2571 \def\eqldimensions@forcall{#1}%
2572 \expandafter\eqldimensions@forstep\eqldimensions@tab
2573 }

```

ldimensions@forstep

```

2574 \def\eqldimensions@forstep\eqldrow@#1\relax#2,#3;%
2575 \eqldrow@#1\relax
2576 \ifnum\eqldrow@=\z@\else
2577 #3%
2578 \def\eqldimensions@cells{#2}%
2579 \eqldimensions@forcall
2580 \expandafter\eqldimensions@forstep
2581 \fi
2582 }

```

\eqldimensions@get

```

2583 \def\eqldimensions@get#1{%
2584 \eqldrow@#1\relax
2585 \expandafter\eqldimensions@getdef\expandafter{\the\eqldrow@}%
2586 \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil
2587 }

```

eqldimensions@getdef

```

2588 \def\eqldimensions@getdef#1{%
2589 \def\eqldimensions@getparse
2590 ##1\eqldrow@#1\relax##2,##3;##4\@nil{%
2591 ##3%
2592 \def\eqldimensions@cells{##2}%
2593 }%
2594 }

```

\eqldcolwidth@tab

```

2595 \let\eqldcolwidth@tab\@empty

```

\eqldcolwidth@get

```

2596 \def\eqldcolwidth@get#1{%
2597 \ifcase\expandafter#1\eqldcolwidth@tab\else\z@\fi
2598 }

```

```

\eql@colwidth@save

2599 \def\eql@colwidth@save#1{%
2600   \edef\eql@colwidth@tab{%
2601     \noexpand\or\the#1%
2602     \unexpanded\expandafter{\eql@colwidth@tab}%
2603   }%
2604 }

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

2605 \def\eql@dimensions@calc{%
2606   \eql@column@\z@
2607   \eql@line@pos@\z@
2608   \eql@line@possep@\z@
2609   \eql@line@avail@\eql@totalwidth@
2610   \eql@line@availsep@\eql@intercolumns@
2611   \eql@line@width@\z@
2612   \eql@line@widthsep@\z@
2613   \let\eql@dimensions@cellcall\eql@dimensions@calc@call
2614   \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
2615   \eql@dimensions@cells
2616 }

ensions@calc@callsep Callback for each intercolumn space.

2617 \def\eql@dimensions@calc@callsep{%
2618   \advance\eql@line@possep@\@ne
2619 }%

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?!)

2620 \def\eql@dimensions@calc@call{%
2621   \advance\eql@column@\@ne
2622   \ifnum\eql@totalcolumns@=\@ne
2623     \dimen@\eql@totalwidth@
2624   \else
2625     \dimen@\eql@colwidth@get\eql@column@\relax
2626   \fi
2627   \ifdim\eql@cellwidth@>\z@
2628     \ifdim\eql@line@width@=\z@
2629       \eql@line@avail@\eql@line@pos@
2630       \eql@line@availsep@\eql@line@possep@
2631       \ifcase\eql@shape@pos@
2632       \or
2633         \advance\eql@line@avail@\dimexpr
2634           (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2635       \or
2636         \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
2637       \fi
2638       \advance\eql@line@avail@\eql@shape@amount@
2639     \fi

```

```

2640 \eqlline@width@eqlline@pos@
2641 \eqlline@widthsep@eqlline@possep@
2642 \ifcase\eql@shape@pos@
2643 \advance\eqlline@width@eql@cellwidth@
2644 \or
2645 \advance\eqlline@width@dimexpr
2646 (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@relax
2647 \or
2648 \advance\eqlline@width@dimen@
2649 \fi
2650 \advance\eqlline@width@eql@shape@amount@
2651 \fi
2652 \advance\eqlline@pos@dimen@
2653 }

```

I.4 Best Line Selection

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

```
2654 \let\eq@numbering@best@auto\eq@false
```

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

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

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

```

2655 \newcount\eq@numbering@best@row@
2656 \newdimen\eq@numbering@best@space@
2657 \let\eq@numbering@best@use\eq@false

```

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

```

2658 \def\eq@numbering@best@find{%
2659 \eq@numbering@best@row@z@
2660 \eq@numbering@best@space@z@
2661 \eq@dimensions@for{%
2662 \eq@dimensions@calc
2663 \ifdefined\eq@tagsleft
2664 \dimen@\eqlline@avail@
2665 \else
2666 \dimen@\dimexpr\eq@totalwidth@-\eqlline@width@relax
2667 \fi
2668 \ifdim\dimen@>\eq@numbering@best@space@
2669 \eq@numbering@best@row@\eq@row@
2670 \eq@numbering@best@space@\dimen@
2671 \fi
2672 }%
2673 \ifnum\eq@numbering@best@row@>z@
2674 \eq@tagpos@row@\eq@numbering@best@row@
2675 \let\eq@tagpos@continuous\eq@false
2676 \eq@tagpos@prevrow@z@
2677 \fi
2678 }

```

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

```

2679 \def\eq@numbering@best@test#1{%
2680 \eq@dimensions@get#1%
2681 \eq@dimensions@calc
2682 \ifdefined\eq@tagsleft
2683 \dimen@\dimexpr\eqlline@avail@

```

```

2684      +\eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax
2685 \else
2686   \dimen@\dimexpr\displaywidth-\eql@line@width@
2687   -\eql@marginleft@-\eql@line@widthsep@\eql@colsep@\relax
2688 \fi
2689 \ifdim\dimen@<\eql@tagwidth@block@
2690   \let\eql@numbering@best@use\eql@true
2691 \fi
2692 }

```

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

```

2693 \def\eql@numbering@best@eval{%
2694   \ifdefined\eql@numbering@best@auto
2695     \ifdefined\eql@numbering@best@use\else
2696       \ifdefined\eql@numbering@multi\else
2697         \ifnum\eql@tagpos@row@>\z@
2698           \eql@numbering@best@test\eql@tagpos@row@
2699         \fi
2700         \ifnum\eql@tagpos@prevrow@>\z@
2701           \eql@numbering@best@test\eql@tagpos@prevrow@
2702         \fi
2703       \fi
2704     \fi
2705   \fi
2706   \ifdefined\eql@numbering@best@use
2707     \eql@numbering@best@find
2708   \fi
2709 }

```

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`

```

2710 \def\eql@adjust@calc@tagmargin{%
2711   \ifdefined\eql@tagmargin@val
2712     \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2713   \else
2714     \eql@tagmargin@\eql@tagwidth@max@
2715     \ifdim\eql@tagmargin@>\z@
2716       \advance\eql@tagmargin@-\eql@tagsepmin@
2717     \fi
2718   \fi

2719   \dimen@\eql@tagrows@p@
2720   \ifnum\eql@totalrows@=\@ne
2721     \ifnum\eql@tagrows@=\@ne
2722       \advance\dimen@1sp\relax
2723     \fi
2724   \fi
2725   \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
2726     \eql@tagmargin@\z@
2727   \fi

2728   \@tempdima\dimexpr\displaywidth

```

```

2729     -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
2730 \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
2731 \ifdim\@tempdimb>\z@
2732     \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
2733         \eql@tagmargin@\z@
2734     \fi
2735 \fi
2736 }

```

I.6 Single Column

`\eql@adjust@calc@lines`

```

2737 \def\eql@adjust@calc@lines{%
2738     \eql@totalcolumns@\@ne
2739     \eql@intercolumns@\z@
2740     \eql@colsep@\z@
2741     \ifdefined\eql@layoutleft
2742         \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2743         \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2744         \ifdim\eql@marginleft@<\eql@marginleft@min@
2745             \eql@marginleft@\eql@marginleft@min@
2746         \fi
2747         \dimen@\glueexpr\eql@layoutleftmarginmax\relax
2748         \ifdim\eql@marginleft@>\dimen@
2749             \eql@marginleft@\dimen@
2750         \fi
2751         \eql@marginright@\z@
2752         \eql@centeroffset@\z@
2753     \else
2754         \eql@adjust@calc@tagmargin
2755         \ifdefined\eql@paddingleft@val
2756             \eql@marginleft@\dimexpr
2757                 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2758                 -\glueexpr\eql@paddingleft@val\relax\relax
2759             \ifdim\eql@marginleft@<\z@
2760                 \eql@marginleft@\z@
2761             \fi
2762         \else
2763             \eql@marginleft@\z@
2764         \fi
2765         \ifdefined\eql@paddingright@val
2766             \eql@marginright@\dimexpr
2767                 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2768                 -\glueexpr\eql@paddingright@val\relax\relax
2769             \ifdim\eql@marginright@<\z@
2770                 \eql@marginright@\z@
2771             \fi
2772         \else
2773             \eql@marginright@\z@
2774         \fi
2775         \ifdim\eql@tagmargin@>\z@
2776             \ifdefined\eql@tagsleft
2777                 \ifdim\eql@marginleft@<\eql@tagsepmin@
2778                     \eql@marginleft@\eql@tagsepmin@
2779                 \fi
2780             \advance\eql@marginleft@\eql@tagmargin@
2781             \advance\eql@centeroffset@\eql@tagmargin@

```



```

2782     \else
2783     \ifdim\eq@marginright@<\eq@tagsepmin@
2784     \eq@marginright@\eq@tagsepmin@
2785     \fi
2786     \advance\eq@marginright@\eq@tagmargin@
2787     \advance\eq@centeroffset@-\eq@tagmargin@
2788     \fi
2789     \fi
2790     \eq@marginleft@min@z@
2791     \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@
2792     \ifdefined\eq@tagsleft+ \else-\fi\eq@tagmargin@\relax
2793     \fi

2794     \eq@totalwidth@\dimexpr\displaywidth
2795     -\eq@marginleft@-\eq@marginright@\relax
2796 }

```

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:

```

2797 \def\eq@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:

```

2798   \ifnum\eq@totalcolumns@<\thr@@
2799   \eq@totalcolumns@\tw@
2800   \let\eq@columns@fulllength\eq@false
2801   \fi

```

Determine the number of intercolumn spaces \eq@intercolumns@:

```

2802   \eq@intercolumns@\numexpr(\eq@totalcolumns@-\tw@)/\tw@\relax

```

Evaluate the minimum intercolumn space which we will need often:

```

2803   \eq@colsepmin@\glueexpr\eq@colsepmin@val\relax

```

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

```

2804   \ifdefined\eq@layoutleft
2805   \eq@marginleft@\glueexpr\eq@layoutleftmargin\relax
2806   \eq@marginleft@min@\glueexpr\eq@layoutleftmarginmin\relax
2807   \ifdim\eq@marginleft@<\eq@marginleft@min@
2808   \eq@marginleft@\eq@marginleft@min@
2809   \fi
2810   \else

```

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

```

2811   \eq@adjust@calc@tagmargin
2812   \ifdefined\eq@columns@fulllength

```

```

2813     \ifdim\eql@tagmargin@>\z@
2814     \advance\eql@tagmargin@\eql@tagsepmin@
2815     \fi
2816   \fi
2817   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2818     -\eql@intercolumns@\eql@colsepmin@\relax
2819     \eql@tagmargin@\z@
2820   \fi
2821   \eql@marginleft@min@\z@
2822 \fi

```

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

```

2823 \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:

```

2824   \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2825   \ifdefined\eql@layoutleft
2826     \advance\eql@colsep@-\eql@marginleft@
2827   \else
2828     \advance\eql@colsep@-\eql@tagmargin@
2829   \fi
2830   \count@\eql@intercolumns@
2831   \ifdefined\eql@columns@fulllength\else
2832     \ifdefined\eql@layoutleft
2833       \advance\count@\@ne
2834     \else
2835       \advance\count@\tw@
2836     \fi
2837   \fi
2838   \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:

```

2839   \ifdim\eql@colsep@<\eql@colsepmin@
2840     \eql@colsep@\eql@colsepmin@
2841   \else
2842     \ifdefined\eql@columns@fulllength\else
2843       \dimen@\glueexpr\eql@colsepmax@val\relax
2844       \ifdim\eql@colsep@>\dimen@
2845         \eql@colsep@\dimen@
2846       \fi
2847     \fi
2848   \fi
2849 \else

```

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

```

2850   \eql@colsep@\eql@colsepmin@
2851 \fi

```

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

```

2852 \ifdefined\eql@layoutleft

```

Set the default value:

```

2853   \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:

```

2854     \dimen@\dimexpr\displaywidth-\eql@totalwidth@
2855         -\eql@intercolumns@\eql@colsep@\relax
2856     \ifdim\dimen@<\eql@marginleft@
2857         \ifdim\dimen@<\eql@marginleft@min@
2858             \eql@marginleft@\eql@marginleft@min@
2859         \else
2860             \eql@marginleft@\dimen@
2861         \fi
2862     \fi
2863 \fi
2864 \else

```

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

```

2865     \ifdefined\eql@columns@fulllength
2866         \eql@marginleft@z@

```

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

```

2867 \else
2868     \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@
2869         -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
2870     \ifdim\eql@marginleft@<z@
2871         \eql@marginleft@z@
2872     \fi
2873 \fi

```

Add tag margin in case of left tags:

```

2874     \ifdefined\eql@tagsleft
2875         \advance\eql@marginleft@\eql@tagmargin@
2876     \fi
2877 \fi

```

Find the best row for tag placement:

```

2878     \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:

```

2879 \ifdefined\eql@tagsleft
2880     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
2881 \else
2882     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2883 \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

```

2884 \ifdefined\eql@numbering@multi
2885     \eql@dimensions@for{%
2886         \ifdim\eql@tagwidth@>z@
2887             \eql@dimensions@calc
2888             \eql@adjust@columns@test

```

```

2889     \fi
2890   }%
2891 \else
2892   \ifnum\eql@tagpos@row@>\z@
2893     \ifnum\eql@tagpos@row@>\eql@totalrows@\else
2894       \eql@dimensions@get\eql@tagpos@row@
2895       \eql@tagwidth@\eql@tagwidth@block@
2896       \eql@dimensions@calc
2897       \eql@adjust@columns@test
2898     \fi
2899   \fi
2900   \ifnum\eql@tagpos@prevrow@>\z@
2901     \eql@dimensions@get\eql@tagpos@prevrow@
2902     \eql@tagwidth@\eql@tagwidth@block@
2903     \eql@dimensions@calc
2904     \eql@adjust@columns@test
2905   \fi
2906 \fi

```

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

```

2907 \advance\eql@totalwidth@\dimexpr
2908   \eql@intercolumns@\eql@colsep@+\eql@marginleft@\relax
2909 }

```

Placement for Right Tags.

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

```

2910 \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?

```

2911 \@tempdima\dimexpr\displaywidth-\eql@line@width@-\eql@tagwidth@\relax

```

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

```

2912 \ifdim\@tempdima<\dimexpr
2913   \eql@marginleft@+\eql@line@widthsep@\eql@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).

```

2914 \ifdim\@tempdima<\dimexpr
2915   \eql@marginleft@min@+\eql@line@widthsep@\eql@colsepmin@\relax\else

```

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

```

2916   \eql@adjust@columns@modify@tagsright
2917 \fi
2918 \fi
2919 }

```

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

```

2920 \def\eql@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.

```

2921 \ifnum\eq@line@widthsep@>\z@
2922   \dimen@ \@tempdima
2923   \count@ \eq@line@widthsep@
2924   \ifdefined\eq@layoutleft
2925     \advance\dimen@-\eq@marginleft@
2926   \else
2927     \ifdefined\eq@columns@fulllength\else
2928       \advance\count@\@ne
2929     \fi
2930   \fi
2931   \divide\dimen@\count@

```

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

```

2932   \ifdim\dimen@<\eq@colsep@
2933     \ifdim\dimen@<\eq@colsepmin@
2934       \eq@colsep@ \eq@colsepmin@
2935     \else
2936       \eq@colsep@ \dimen@
2937     \fi
2938   \fi
2939 \fi

```

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

```

2940 \dimen@ \dimexpr \@tempdima-\eq@line@widthsep@ \eq@colsep@ \relax
2941 \ifdim\eq@marginleft@>\dimen@
2942   \eq@marginleft@ \dimen@
2943 \fi
2944 }

```

Placement for Left Tags.

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

```

2945 \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.

```

2946 \count@ \numexpr \eq@intercolumns@-\eq@line@availsep@ \relax
2947 \@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.

```

2948 \ifdim \@tempdima > \dimexpr
2949   \eq@marginleft@ + \eq@line@availsep@ \eq@colsep@ \relax

```

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

```

2950   \ifdim \eq@tagwidth@ < %

```

```

2951      \ifdefined\eq@layoutleft
2952      \glueexpr\eq@layoutleftmarginmax\relax
2953      \else
2954      \displaywidth
2955      \fi

```

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

```

2956      \ifdim\@tempdima>\dimexpr
2957      \displaywidth-\eq@totalwidth@-\count@eq@colsepmin@\relax\else

```

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

```

2958      \eq@adjust@columns@modify@tagsleft
2959      \fi
2960      \fi
2961      \fi
2962 }

```

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

```

2963 \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.

```

2964 \ifnum\count@>\z@
2965 \dimen@dimexpr\displaywidth-\eq@totalwidth@-\@tempdima\relax
2966 \ifdefined\eq@columns@fulllength\else
2967 \advance\count@1
2968 \fi
2969 \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.

```

2970 \ifdim\dimen@<\eq@colsep@
2971 \ifdim\dimen@<\eq@colsepmin@
2972 \dimen@eq@colsepmin@
2973 \fi
2974 \advance\dimen@-\eq@colsep@
2975 \advance\eq@marginleft@-\eq@intercolumns@\dimen@
2976 \advance\eq@colsep@\dimen@
2977 \fi
2978 \fi

```

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

```

2979 \dimen@dimexpr\@tempdima-\eq@line@availsep@\eq@colsep@\relax
2980 \ifdim\eq@marginleft@<\dimen@
2981 \eq@marginleft@\dimen@
2982 \fi
2983 }

```

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:

```
2984 \ifdefined\inf@bad\else%
2985   \newcount\inf@bad
2986   \inf@bad1000000\relax
2987 \fi
```

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

```
2988 \let\eq@restore@hfuzz\@empty
2989 \def\eq@save@hfuzz{\edef\eq@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

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

```
2990 \newcount\eq@alignbadness@
2991 \newcount\eq@tagbadness@
2992 \newcount\eq@arrange@badness@
2993 \eq@alignbadness@\inf@bad
2994 \eq@tagbadness@\inf@bad
```

J.2 Arrangement Methods

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

```
2995 \def\eq@arrange@try#1{%
2996   \ifdim#1>\dimexpr\displaywidth-\eq@cellwidth@\relax
2997     \setbox\eq@cellbox@\hbox to\displaywidth{%
2998       \unhbox\eq@cellbox@\unkern\kern#1}%
2999     \eq@arrange@badness@\badness
3000   \else
3001     \eq@arrange@badness@\m@ne
3002   \fi
3003 }
```

`\eq@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

```
3004 \def\eq@arrange@print#1#2{%
3005   \eq@restore@hfuzz
3006   \if@eqnsw
3007     \ifdefined\eq@tagsleft
3008       \eq@tagbox@print@tagsleft
3009     \fi
```

```

3010 \fi
3011 \hbox to\displaywidth{%
3012   #1%
3013   \unhbox\eql@cellbox@\unkern
3014   #2%
3015   \eql@tagging@mathaddlast
3016 }%
3017 \if@eqnsw
3018   \ifdefined\eql@tagsleft\else
3019     \eql@tagbox@print@tagsright
3020   \fi
3021 \fi
3022 }

```

`\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:

```

3023 \def\eql@arrange@print@alignleft#1#2{%
3024   \eql@display@firstavail@set{\dimexpr#1\relax}%
3025   \eql@arrange@print{\kern#1}{\kern#2}%
3026 }

3027 \def\eql@arrange@print@alignright#1#2{%
3028   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3029   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3030 }

3031 \def\eql@arrange@print@aligncenter#1{%
3032   \eql@display@firstavail@set{\dimexpr
3033     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3034   \ifdim#1>\z@
3035     \eql@arrange@print{\kern#1\hfil}{}%
3036   \else
3037     \eql@arrange@print{\hfil}{\kern-#1}%
3038   \fi
3039 }

```

`\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:

```

3040 \def\eql@arrange@init{%
3041   \eql@save@hfuzz
3042   \hfuzz\maxdimen
3043   \eql@shape@select
3044 }

```

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

```

3045 \def\eql@arrange@print@line{%
3046   \eql@tagging@tagaddbox
3047   \csname eql@arrange%
3048     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3049     @init\endcsname
3050   \csname eql@arrange%
3051     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3052     @\ifdefined\eql@tagpos@reserve

```



```

3053 \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3054 notag\fi\endcsname
3055 }

```

J.3 Central Alignment

TODO: describe

```

3056 \def\eql@arrange@aligncenter@init{%
3057 \eql@tagging@aligncenter
3058 \eql@line@offset@<\dimexpr\tw@<\eql@shape@amount@
3059 +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@>\relax
3060 }

```

TODO: describe

```

3061 \def\eql@arrange@aligncenter@notag{%
3062 \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3063 \ifdim\eql@line@offset@<\eql@marginleft@min@
3064 \dimexpr\tw@<\eql@marginleft@min@-\eql@line@offset@>\relax
3065 \else
3066 \eql@line@offset@
3067 \fi
3068 \eql@arrange@print@aligncenter\eql@line@offset@
3069 \else
3070 \ifdim\eql@line@offset@<\eql@marginleft@min@
3071 \eql@arrange@print@alignleft\eql@marginleft@min@>\z@
3072 \else
3073 \eql@arrange@print@alignright\eql@marginleft@min@>\z@
3074 \fi
3075 \fi
3076 }

```

TODO: describe

```

3077 \def\eql@arrange@aligncenter@tagsright{%
3078 \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3079 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@>\relax
3080 \dimexpr\tw@<\eql@marginleft@min@-\eql@line@offset@>\relax
3081 \else
3082 \dimexpr\tw@<\eql@tagwidth@+\eql@line@offset@>\relax
3083 \fi
3084 \eql@arrange@print@aligncenter\eql@line@offset@
3085 \else
3086 \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@>\relax}%
3087 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3088 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@>\relax
3089 \eql@arrange@print@alignleft\eql@marginleft@min@>\eql@tagwidth@
3090 \else
3091 \eql@arrange@print@alignright\eql@marginleft@min@>\eql@tagwidth@
3092 \fi
3093 \else
3094 \let\eql@tagpos@reserve\eql@false
3095 \eql@arrange@aligncenter@notag
3096 \fi
3097 \fi
3098 }

```

```

3099 \def\eql@arrange@aligncenter@tagsleft{%

```

```

3100 \ifdim\eq\tagwidth@>\eq\marginleft@min@
3101 \ifdim\dimexpr\displaywidth-\eq\cellwidth@>\relax%
3102 \ifdim\eq\line@offset@<\eq\tagwidth@
3103 \dimexpr\tw@\eq\tagwidth@-\eq\line@offset@>\relax
3104 \else
3105 \eq\line@offset@
3106 \fi
3107 \eq\arrange@print@aligncenter\eq\line@offset@
3108 \else
3109 \eq\arrange@try\eq\tagwidth@
3110 \ifnum\eq\arrange@badness@<\eq\tagbadness@
3111 \ifdim\eq\line@offset@<\eq\tagwidth@
3112 \eq\arrange@print@alignleft\eq\tagwidth@\z@
3113 \else
3114 \eq\arrange@print@alignright\eq\tagwidth@\z@
3115 \fi
3116 \else
3117 \let\eq\tagpos@reserve\eq>false
3118 \eq\arrange@aligncenter@notag
3119 \fi
3120 \fi
3121 \else
3122 \eq\arrange@aligncenter@notag
3123 \fi
3124 }

```

J.4 Left Alignment

```

3125 \def\eq\arrange@alignleft@init{%
3126 \eq\tagging@alignleft
3127 \eq\line@offset@\dimexpr\eq\marginleft@+\eq\shape@amount@>\relax
3128 \ifdim\eq\line@offset@<\eq\marginleft@min@
3129 \eq\line@offset@\eq\marginleft@min@
3130 \fi
3131 }

3132 \def\eq\arrange@alignleft@notag{%
3133 \ifdim\eq\line@offset@>\eq\marginleft@min@
3134 \eq\arrange@try\eq\line@offset@
3135 \ifnum\eq\arrange@badness@<\eq\alignbadness@
3136 \eq\arrange@print@alignleft\eq\line@offset@\z@
3137 \else
3138 \eq\arrange@print@alignright\eq\marginleft@min@\z@
3139 \fi
3140 \else
3141 \eq\arrange@print@alignleft\eq\marginleft@min@\z@
3142 \fi
3143 }

3144 \def\eq\arrange@alignleft@tagsright{%
3145 \eq\arrange@try{\dimexpr\eq\line@offset@+\eq\tagwidth@>\relax}%
3146 \ifnum\eq\arrange@badness@<\eq\alignbadness@
3147 \eq\arrange@print@alignleft\eq\line@offset@\eq\tagwidth@
3148 \else
3149 \ifdim\eq\line@offset@>\eq\marginleft@min@
3150 \eq\arrange@try{\dimexpr\eq\marginleft@min@+\eq\tagwidth@>\relax}%
3151 \fi
3152 \ifnum\eq\arrange@badness@<\eq\tagbadness@
3153 \eq\arrange@print@alignright\eq\marginleft@min@\eq\tagwidth@

```

```

3154     \else
3155         \let\eql@tagpos@reserve\eql@false
3156         \eql@arrange@alignleft@notag
3157     \fi
3158 \fi
3159 }

3160 \def\eql@arrange@alignleft@tagsleft{%
3161     \ifdim\eql@tagwidth@>\eql@marginleft@min@
3162         \ifdim\eql@line@offset@>\eql@tagwidth@
3163             \eql@arrange@try\eql@line@offset@
3164             \ifnum\eql@arrange@badness@<\eql@alignbadness@
3165                 \eql@arrange@print@alignleft\eql@line@offset@z@
3166             \else
3167                 \eql@arrange@try\eql@tagwidth@
3168                 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3169                     \eql@arrange@print@alignright\eql@tagwidth@z@
3170                 \else
3171                     \let\eql@tagpos@reserve\eql@false
3172                     \eql@arrange@print@alignright\eql@marginleft@min@z@
3173                 \fi
3174             \fi
3175         \else
3176             \eql@arrange@try\eql@tagwidth@
3177             \ifnum\eql@arrange@badness@<\eql@tagbadness@
3178                 \eql@arrange@print@alignleft\eql@tagwidth@z@
3179             \else
3180                 \let\eql@tagpos@reserve\eql@false
3181                 \eql@arrange@alignleft@notag
3182             \fi
3183         \fi
3184     \else
3185         \eql@arrange@alignleft@notag
3186     \fi
3187 }

```

J.5 Right Alignment

```

3188 \def\eql@arrange@alignright@init{%
3189     \eql@tagging@alignright
3190     \eql@line@offset@dimexpr\eql@marginright@-\eql@shape@amount@relax
3191     \ifdim\eql@line@offset@<z@
3192         \eql@line@offset@z@
3193     \fi
3194 }

```

TODO: describe

```

3195 \def\eql@arrange@alignright@notag{%
3196     \ifdim\eql@line@offset@>z@
3197         \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@relax}%
3198         \ifnum\eql@arrange@badness@<\eql@alignbadness@
3199             \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3200         \else
3201             \eql@arrange@print@alignleft\eql@marginleft@min@z@
3202         \fi
3203     \else
3204         \eql@arrange@print@alignright\eql@marginleft@min@z@
3205     \fi
3206 }

```

TODO: describe

```
3207 \def\eql@arrange@alignright@tagsright{%
3208   \ifdim\eql@line@offset@>\eql@tagwidth@
3209     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3210     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3211       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3212     \else
3213       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3214       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3215         \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3216       \else
3217         \let\eql@tagpos@reserve\eql@false
3218         \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3219       \fi
3220     \fi
3221 \else
3222   \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3223   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3224     \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3225   \else
3226     \let\eql@tagpos@reserve\eql@false
3227     \eql@arrange@alignright@notag
3228   \fi
3229 \fi
3230 }
```

TODO: describe

```
3231 \def\eql@arrange@alignright@tagsleft{%
3232   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3233     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3234     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3235       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3236     \else
3237       \ifdim\eql@line@offset@>\z@
3238         \eql@arrange@try\eql@tagwidth@
3239       \fi
3240       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3241         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3242       \else
3243         \let\eql@tagpos@reserve\eql@false
3244         \eql@arrange@alignright@notag
3245       \fi
3246     \fi
3247 \else
3248   \eql@arrange@alignright@notag
3249 \fi
3250 }
```

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

```

3251 \protected\def\eql@box@cr{%
3252   \eql@ampprotecttwo{\eql@ifnextchar@tight[]\eql@box@cr@skip\eql@box@cr@
3253 }
3254 \def\eql@box@cr@{%
3255   \eql@punct@apply@line
3256   \eql@hook@lineout
3257   \eql@box@lastcell
3258   \cr
3259 }
3260 \def\eql@box@cr@skip[#1]{%
3261   \eql@box@cr@
3262   \noalign{%
3263     \vskip\glueexpr#1\relax
3264   }%
3265 }
```

K.2 Stacked Mode

```

3266 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}
```

TODO: templates

```

3267 \def\eql@box@open@stacked{%
3268   \eql@shape@align@enable
3269   \let\eql@box@lastcell\eql@box@lastcell@stacked
3270   \everycr{\noalign{%
3271 (dev)\eql@dev{starting line \the\eql@row@}%
3272   \global\advance\eql@row@\@ne
3273   }}%
3274   \tabskip\z@skip
3275   \halign\bgroup
3276     &%
3277     \global\let\eql@cell@container\@empty
3278     \setbox\eql@cellbox@\hbox{%
3279       \eql@strut@cell
3280       \@lign
3281       $\m@th\displaystyle
3282       \eql@hook@colin
3283       ##%
3284       \eql@punct@apply@col
3285       \eql@hook@colout
3286       \eql@tagging@mathsave
3287       $%
3288       \eql@tagging@mathaddlast
3289     }%
3290     \ifdefined\eql@shape@lastrow
3291       \eql@totalrows@\eql@row@
3292     \fi
3293     \eql@shape@eval
3294     \eql@cell@container
3295     \ifdefined\eql@frame@cmd
3296       \ifcase\eql@shape@pos@
3297         \eql@frame@measure
3298         \advance\eql@shape@amount@-\eql@frame@margin@
3299       \or\or
```

```

3300         \eql@frame@measure
3301         \advance\eql@shape@amount@+\eql@frame@margin@
3302     \fi
3303     \eql@frame@print
3304 \fi
3305 \ifcase\eql@shape@pos@
3306     \kern\eql@shape@amount@
3307     \box\eql@cellbox@
3308     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3309     -\eql@shape@amount@+\@flushglue\relax
3310     \eql@tagging@alignleft
3311 \or
3312     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3313     \box\eql@cellbox@
3314     \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3315     \eql@tagging@aligncenter
3316 \or
3317     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3318     +\eql@shape@amount@+\@flushglue\relax
3319     \box\eql@cellbox@
3320     \kern-\eql@shape@amount@
3321     \eql@tagging@alignright
3322 \fi
3323     \tabskip\eql@colsep@\relax
3324 \crrc
3325 \noalign{%
3326     \global\let\eql@shape@lastrow\eql@false
3327     \eql@hook@blockbefore
3328 }%
3329 \eql@hook@blockin
3330 }
3331 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

K.3 Aligned Mode

```

3332 \def\eql@box@lastcell@odd{%
3333     &\omit
3334     \eql@prevwidth@\wd\eql@cellbox@
3335     \let\eql@frame@cmd\eql@frame@prevcmd
3336     \ifdefined\eql@frame@cmd
3337         \eql@frame@measure
3338         \advance\eql@prevwidth@\eql@frame@margin@
3339         \eql@frame@print
3340     \fi
3341     \kern-\eql@prevwidth@
3342     \unhbox\eql@cellbox@
3343     \hfil
3344     &\omit\kern-\eql@colsep@
3345 }%
3346 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}
3347 \def\eql@box@open@aligned{%
3348 % \TODO templates
3349     \eql@shape@align@disable
3350     \let\eql@box@lastcell\@empty
3351     \everycr{\noalign{%
3352 (dev)\eql@dev{starting new line}%
3353     }}%
3354     \tabskip\z@skip

```

```

3355 \halign\bgroup
3356   &%
3357   \let\eql@box@lastcell\eql@box@lastcell@odd
3358   \global\let\eql@cell@container\@empty
3359   \global\setbox\eql@cellbox@\hbox{%
3360     \eql@strut@cell
3361     \@lign
3362     $\m@th\displaystyle
3363     \eql@hook@colin
3364     ##%
3365     \eql@class@innerleft
3366     \eql@hook@innerleft
3367     \eql@tagging@mathsave
3368     $%
3369     \eql@tagging@mathaddlast
3370   }%
3371   \eql@cell@container
3372   \hfil
3373   \kern\wd\eql@cellbox@
3374   \ifdefined\eql@frame@cmd
3375     \eql@frame@measure
3376     \kern\eql@frame@margin@
3377   \fi
3378   \global\let\eql@frame@prevcmd\eql@frame@cmd
3379   \tabskip\z@skip
3380   &%
3381   \eql@prevwidth@\wd\eql@cellbox@
3382   \let\eql@box@lastcell\eql@box@lastcell@even
3383   \let\eql@frame@cmd\eql@frame@prevcmd
3384   \global\let\eql@cell@container\@empty
3385   \setbox\eql@cellbox@\hbox{%
3386     \unhbox\eql@cellbox@
3387     \eql@strut@cell
3388     \@lign
3389     $\m@th\displaystyle
3390     \eql@hook@innerright
3391     \eql@class@innerright@sel
3392     ##%
3393     \eql@punct@apply@col
3394     \eql@hook@colout
3395     \eql@tagging@mathsave
3396     $%
3397     \eql@tagging@mathaddlast
3398   }%
3399   \eql@cell@container
3400   \ifdefined\eql@frame@cmd
3401     \eql@frame@measure
3402     \advance\eql@prevwidth@\eql@frame@margin@
3403     \eql@frame@print
3404   \fi
3405   \kern-\eql@prevwidth@
3406   \unhbox\eql@cellbox@
3407   \hfil
3408   \tabskip\eql@colsep@\relax
3409 \crrc
3410 \noalign{%
3411   \eql@hook@blockbefore
3412 }%

```

```

3413 \eql@hook@blockin
3414 }
3415 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

K.4 Main

```

3416 \let\eql@box@box\vcenter
3417 \let\eql@box@open\@undefined
3418 \let\eql@box@frame\@firstofone
3419 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

```

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

```

3420 \def\eql@box@close{%
3421   \ifvmode\else
3422     \global\let\eql@shape@lastrow\eql@true
3423     \eql@punct@apply@block
3424     \eql@box@cr@
3425   \fi
3426   \noalign{%
3427     \eql@hook@blockafter
3428     \global\let\eql@shape@lastrow\eql@false
3429   }%
3430   \eql@tagging@tablesaverinner
3431 \egroup
3432 }

```

`\eql@box@vcenter`

```

3433 \def\eql@box@vcenter#1{%
3434   \ifmmode
3435     \vcenter{#1}%
3436   \else
3437     $\m@th\vcenter{#1}$%
3438   \fi
3439 }

```

`\eql@box@start`

```

3440 \let\eql@box@endmath\eql@false
3441 \def\eql@box@start{%
3442   \relax
3443   \ifmmode
3444     \let\eql@box@endmath\eql@false
3445   \else
3446     \let\eql@box@endmath\eql@true
3447     \expandafter$%$
3448   \fi
3449   \eql@box@processopt
3450   \eql@stack@save@box
3451   \let\eql@frame@cmd\@undefined
3452   \let\eql@layoutleft\eql@false
3453   \eql@row@z@
3454   \eql@totalrows@\@M
3455   \eql@shape@select
3456   \setbox\z@\ifx\eql@box@box\vcenter

```



```

3457 \expandafter\ vbox
3458 \else
3459 \expandafter\ eql@box@box
3460 \fi\bgroup
3461 \eql@display@nest
3462 \let\\eql@box@cr
3463 \eql@spread@set
3464 \eql@strut@make
3465 \eql@box@open
3466 }

```

`\eql@box@end`

```

3467 \def\eql@box@end{%
3468 \eql@box@close
3469 \egroup
3470 \eql@box@frame{%
3471 \ifdefined\eql@display@marginleft
3472 \hskip\glueexpr\eql@display@marginleft\relax
3473 \fi
3474 \ifx\eql@box@box\vcenter
3475 \eql@box@vcenter{\unvbox\z@}%
3476 \else
3477 \box\z@
3478 \fi
3479 \eql@tagging@tableaddinner
3480 \ifdefined\eql@display@marginright
3481 \hskip\glueexpr\eql@display@marginright\relax
3482 \fi
3483 }%
3484 \eql@stack@restore
3485 \ifdefined\eql@box@endmath
3486 \expandafter$%$
3487 \fi
3488 }

```

K.5 Environment

`equationsbox` (*env.*)

```

3489 \newenvironment{equationsbox}{%
3490 (dev)\eql@dev@enterenv
3491 \eql@ampprotect\eql@box@testall\eql@box@start
3492 }{%
3493 \eql@box@end
3494 (dev)\eql@dev@leaveenv
3495 }

3496 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3497 \def\eql@box@parseopt{%
3498 \ifx\eql@parseopt@token[%]
3499 \let\eql@parseopt@next\eql@parseopt@opt
3500 \fi
3501 \ifx\eql@parseopt@token=%
3502 \let\eql@parseopt@next\eql@parseopt@lines
3503 \fi
3504 \ifx\eql@parseopt@token|
3505 \let\eql@parseopt@next\eql@parseopt@columns

```

```

3506 \fi
3507 }

```

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

```

3508 \def\eql@box@processopt{%
3509   \let\eql@box@frame\@firstofone
3510   \let\eql@display@marginleft\@undefined
3511   \let\eql@display@marginright\@undefined
3512   \eql@nextopt@process{equationsbox}%
3513   \let\eql@punct@block\eql@punct@main
3514   \let\eql@punct@main\relax
3515   \eql@colsep@\glueexpr\eql@box@colsep\relax
3516   \ifdefined\eql@paddingleft@val
3517     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3518   \else
3519     \eql@paddingleft@\z@
3520   \fi
3521   \ifdefined\eql@paddingright@val
3522     \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3523   \else
3524     \eql@paddingright@\z@
3525   \fi
3526   \eql@indent@\glueexpr\eql@indent@val\relax
3527 }

```

L Single-Line Equation

TODO: describe

L.1 Native Mode

```

3528 \def\eql@single@start@native{%
3529   \eql@display@init
3530   \eql@display@print
3531   \let\raisetag\eql@raisetag@default
3532   \eql@shape@align@disable
3533   \eql@hook@eqin
3534 % \mathopen{}}%
3535 }%

```

TODO: describe

```

3536 \def\eql@single@end@native{%
3537 % \mathclose{}}%
3538 \eql@tags@container
3539 \eql@numbering@single@eval
3540 \if@eqnsw
3541   \ifdefined\eql@tagsleft
3542     \leqno
3543   \else
3544     \eqno
3545   \fi
3546   \eql@composetag@print
3547 \fi
3548 \eql@interline@container
3549 \advance\eql@belowspace@\eql@vspaceskip@

```

```

3550 \eqldisplay@container
3551 \eqldisplay@penalty
3552 \eqldisplay@vspace@native
3553 }%

```

L.2 Print

```

3554 \def\eql@single@start@print{%
3555   \eqldisplay@init
3556   \eqldisplay@print
3557   \eql@shape@align@enable

3558   \eql@totalrows@ \@ne
3559   \eql@row@ \@ne
3560   \eql@arrange@init
3561   \global\let\eql@cell@container \@empty

3562   \prevgraf\numexpr\prevgraf+\@ne\relax
3563   \setbox\eql@cellbox@\hbox\bgroup
3564     \eql@restore@hfuzz
3565     \eql@strut@cell
3566     $\m@th\displaystyle%$
3567     \eql@hook@eqin
3568 }

3569 \def\eql@single@end@print{%
3570   \eql@tagging@mathsave
3571   $%$
3572   \hfil
3573   \kern\z@
3574   \egroup
3575   \prevgraf\numexpr\prevgraf-\@ne\relax

3576   \eql@shape@eval
3577   \eql@cell@container

3578   \ifdefined\eql@frame@cmd
3579     \eql@frame@adjust
3580   \fi

3581   \eql@cellwidth@\wd\eql@cellbox@
3582   \eql@line@height@\ht\eql@cellbox@
3583   \eql@line@depth@\dp\eql@cellbox@
3584   \eql@totalwidth@\eql@cellwidth@
3585   \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
3586   \eql@topheight@\eql@line@height@
3587   \eql@bottomdepth@\eql@line@depth@

3588   \eql@tags@container
3589   \eql@numbering@single@eval
3590   \if@eqnsw
3591     \eql@tagbox@make\eql@composetag@print
3592     \eql@tagrows@\@ne
3593     \ifdefined\eql@tagpos@reserve\else
3594       \eql@tagwidth@\z@
3595     \fi
3596     \eql@tagheight@block@\ht\eql@tagbox@
3597     \eql@tagdepth@block@\dp\eql@tagbox@
3598   \else
3599     \eql@numbering@warnunused
3600     \eql@tagwidth@\z@
3601     \eql@tagrows@\z@
3602   \fi

```

```

3603 \eqL@tagwidth@max@\eqL@tagwidth@
3604 \eqL@tagpos@single@eval
3605 \eqL@tagpos@print@line@eval
3606 \eqL@intercolumns@z@
3607 \eqL@adjust@calc@lines
3608 \eqL@display@halign@init{}%
3609 \halign{##\crr
3610   \noalign{\eqL@display@halign@start}%
3611   \eqL@arrange@print@line
3612   \cr
3613   \noalign{\eqL@display@halign@end}%
3614   \eqL@tagging@tablesavelines
3615 }%
3616 \eqL@tagpos@print@line@end
3617 \eqL@display@close
3618 }

```

M Multi-Line with Single Column

TODO: outline sequence of calls

M.1 Measure

TODO: describe

```

3619 \def\eqL@lines@measure@line@begin{%
3620 (dev)\eqL@dev{starting line \the\eqL@row}%
3621 \eqL@numbering@measure@line@begin
3622 \eqL@hook@linein
3623 }

```

TODO: describe

```

3624 \def\eqL@lines@measure@line@end{%
3625 \eqL@punct@apply@line
3626 \eqL@hook@lineout
3627 }

```

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

```

3628 \def\eqL@lines@measure@cell{%
3629 \ifdefined\eqL@frame@cmd
3630 \ifcase\eqL@shape@pos@
3631 \eqL@frame@measure
3632 \advance\eqL@shape@amount@-\eqL@frame@margin@
3633 \or\or
3634 \eqL@frame@measure
3635 \advance\eqL@shape@amount@+\eqL@frame@margin@
3636 \fi
3637 \eqL@frame@print
3638 \fi
3639 \eqL@cellwidth@\wd\eqL@cellbox@
3640 \eqL@line@height@\ht\eqL@cellbox@
3641 \eqL@line@depth@\dp\eqL@cellbox@
3642 \eqL@dimensions@startrow
3643 \eqL@dimensions@savecell

```

```

3644 \kern\eq@cellwidth@
3645 }

```

\eq@lines@measure

```

3646 \def\eq@lines@measure{%
3647 (dev)\eq@dev@enter\eq@lines@measure
3648 \eq@measure@init\eq@lines@measure@line@begin\eq@lines@measure@line@end
3649 \eq@totalrows@ \@M
3650 \eq@shape@select

3651 \setbox\z@\vbox{\measuring@true\halign{%
3652     \global\let\eq@cell@container\@empty
3653     \setbox\eq@cellbox@\hbox{%
3654         \eq@strut@cell
3655         \@lign
3656         $\m@th\displaystyle
3657         \eq@hook@colin
3658         ##%
3659         \eq@punct@apply@col
3660         \eq@hook@colout
3661         $%
3662     }%
3663     \ifdefined\eq@shape@lastrow
3664         \eq@totalrows@\eq@row@
3665     \fi
3666     \eq@shape@eval
3667     \eq@cell@container
3668     \eq@lines@measure@cell
3669     \eq@measure@tag
3670     \eq@measure@endrow
3671 \crr

3672 \noalign{%
3673     \global\let\eq@shape@lastrow\eq@false
3674     \eq@hook@blockbefore
3675 }%
3676 \eq@hook@blockin
3677 \eq@scan@body
3678 \ifvmode\else
3679     \global\let\eq@shape@lastrow\eq@true
3680     \eq@punct@apply@block
3681     \eq@hook@blockout
3682     \eq@display@endline
3683     \cr
3684 \fi
3685 \omit
3686 \cr
3687 \noalign{%
3688     \eq@hook@blockafter
3689     \global\let\eq@shape@lastrow\eq@false
3690 }%
3691 }}%

3692 \eq@measure@close

3693 \setbox\z@\vbox{%
3694     \unvbox\z@
3695     \unpenalty

```

```

3696     \global\setbox\@ne\lastbox
3697 }%
3698 \eql@totalwidth@\wd\@ne

3699 <dev>\eql@dev@leave\eql@lines@measure
3700 }

```

M.2 Column Placement

TODO: describe Find the best row for tag placement:

```

3701 \def\eql@lines@adjust{%
3702   \eql@tagpos@adjust@eval
3703   \eql@adjust@calc@lines
3704   \eql@numbering@best@eval
3705 }

```

M.3 Print

TODO: describe

mes@print@line@begin

```

3706 \def\eql@lines@print@line@begin{%
3707 <dev>\eql@dev{starting line \the\eql@row}%
3708   \eql@numbering@print@line@begin
3709   \eql@hook@linein
3710 }

```

TODO: describe

```

3711 \def\eql@lines@print@line@end{%
3712   \eql@punct@apply@line
3713   \eql@hook@lineout
3714 }

```

TODO: describe

```

3715 \def\eql@lines@print@line@adjust{%
3716   \ifdefined\eql@frame@cmd
3717     \ifcase\eql@shape@pos@
3718       \eql@frame@measure
3719       \advance\eql@shape@amount@-\eql@frame@margin@
3720     \or\or
3721       \eql@frame@measure
3722       \advance\eql@shape@amount@+\eql@frame@margin@
3723     \fi
3724     \eql@frame@adjust
3725   \fi
3726   \eql@cellwidth@\wd\eql@cellbox@
3727   \eql@line@height@\ht\eql@cellbox@
3728   \eql@line@depth@\dp\eql@cellbox@
3729   \eql@numbering@print@line@eval
3730   \if@eqnsw
3731     \eql@tagbox@make\eql@composetag@print
3732   \fi
3733   \eql@tagpos@print@line@eval
3734   \eql@arrange@print@line

```

```

3735 \eql@tagpos@print@line@end
3736 }

```

TODO: describe

```

3737 \def\eql@lines@print{%
3738 (dev)\eql@dev@enter\eql@lines@print
3739 \eql@arrange@init
3740 \eql@display@halign@init\eql@lines@print@line@begin
3741 \eql@display@halign@letcr\eql@lines@print@line@end
3742 \tabskip\z@skip

3743 \halign{%
3744 \global\let\eql@cell@container\@empty
3745 \setbox\eql@cellbox@\hbox{%
3746 \eql@restore@hfuzz
3747 \eql@strut@cell
3748 \@lign
3749 $\m@th\displaystyle
3750 \eql@hook@colin
3751 ##%
3752 \eql@punct@apply@col
3753 \eql@hook@colout
3754 \eql@tagging@mathsave
3755 $%
3756 \hfil
3757 \kern\z@
3758 }%
3759 \eql@shape@eval
3760 \eql@cell@container
3761 \eql@lines@print@line@adjust
3762 \cr

3763 \noalign{%
3764 \eql@display@halign@start
3765 \eql@numbering@print@block@begin
3766 \eql@hook@blockbefore
3767 }%
3768 \eql@hook@blockin
3769 \eql@scan@body
3770 \ifvmode\else
3771 \relax
3772 \eql@punct@apply@block
3773 \eql@hook@blockout
3774 \eql@display@endline
3775 \cr
3776 \fi
3777 \noalign{%
3778 \eql@hook@blockafter
3779 \eql@display@halign@end
3780 (dev)\eql@dev@leave\eql@lines@print
3781 }%
3782 \eql@tagging@tablesavelines
3783 }%
3784 }

```

N Multi-Line with Multiple Columns

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

N.1 Support

TODO: describe

```
\eql@columns@add@amp
@columns@completerow
3785 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3786 \def\eql@columns@completerow{%
3787   \count@\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
3788   \edef\eql@tmp{%
3789     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3790   \eql@tmp
3791 }

3792 \def\eql@columns@overfull{%
3793   \dimen@\eql@line@width@
3794   \advance\dimen@-\hfuzz
3795   \ifdim\dimen@>\displaywidth
3796     \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3797     \wd\z@\z@
3798     \ht\z@\eql@line@height@
3799     \dp\z@\eql@line@depth@
3800     \box\z@
3801   \fi
3802 }
```

N.2 Transpose

TODO: describe

TODO: describe

```
3803 \let\eql@transpose@active\eql@false
3804 \def\eql@transpose@end{\eql@transpose@end}
3805 \def\eql@transpose@skip{&\eqnpunct{}}
3806 \def\eql@transpose@complete{%
3807   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}
```

TODO: describe

```
3808 \def\eql@transpose{%
3809   \eql@totalcolumns@\z@
3810   \eql@totalrows@\z@
3811   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
3812   \eql@scan@reg@{}}%
3813   \eql@row@\z@
3814   \eql@transpose@output@row
3815 }
```

TODO: describe

```
3816 \def\eql@transpose@save@col#1{%
3817   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
3818     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}
```


TODO: describe

```
3819 \def\eql@transpose@scan@col#1\&{%
3820   \def\@tempa{#1}%
3821   \ifx\@tempa\eql@transpose@end\else
3822     \advance\eql@totalcolumns@\@ne
3823     \eql@row@\z@
3824     \let\eql@transpose@data@col\@empty
3825     \eql@transpose@scan@row#1\\eql@transpose@end\\%
3826     \ifnum\eql@row@>\eql@totalrows@
3827       \eql@totalrows@\eql@row@
3828     \fi
3829     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
3830     \expandafter\eql@transpose@scan@col
3831   \fi
3832 }
```

TODO: describe

```
3833 \def\eql@transpose@append@row#1{%
3834   \advance\eql@row@\@ne
3835   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}
```

TODO: describe

```
3836 \def\eql@transpose@scan@row#1\\{%
3837   \def\@tempa{#1}%
3838   \ifx\@tempa\eql@transpose@end\else
3839     \ifx\eql@transpose@active+
3840       \eql@transpose@scan@cell#1\eql@transpose@end&%
3841     \else
3842       \eql@transpose@append@row{#1}%
3843     \fi
3844     \expandafter\eql@transpose@scan@row
3845   \fi
3846 }
```

TODO: describe

```
3847 \def\eql@transpose@scan@cell#1&#2&{%
3848   \def\@tempa{#2}%
3849   \ifx\@tempa\eql@transpose@end
3850     \eql@transpose@append@row{#1}%
3851   \else
3852     \eql@transpose@append@row{#1&#2}%
3853     \expandafter\eql@transpose@scan@cell@next
3854   \fi
3855 }
```

TODO: describe

```
3856 \def\eql@transpose@scan@cell@next#1&{%
3857   \def\@tempa{#1}%
3858   \ifx\@tempa\eql@transpose@end\else
3859     \eql@transpose@append@row{&#1}%
3860     \expandafter\eql@transpose@scan@cell@next
3861   \fi
3862 }
```

TODO: describe

```
3863 \def\eql@transpose@output@row{%
```

```

3864 \ifnum\eqL@row@<\eqL@totalrows@
3865 \advance\eqL@row@\@ne
3866 \eqL@column@\z@
3867 \eqL@transpose@output@col
3868 \ifnum\eqL@row@<\eqL@totalrows@
3869 \eqL@scan@addto\\%
3870 \fi
3871 \expandafter\eqL@transpose@output@row
3872 \fi
3873 }

```

TODO: describe

```

3874 \def\eqL@transpose@output@col{%
3875 \ifnum\eqL@column@<\eqL@totalcolumns@
3876 \advance\eqL@column@\@ne
3877 \csname eqL@transpose@data@col@\the\eqL@column@\endcsname
3878 \expandafter\eqL@scan@addto\expandafter{\eqL@tmp}%
3879 \ifnum\eqL@column@<\eqL@totalcolumns@
3880 \eqL@scan@addto{\eqL@transpose@complete}%
3881 \fi
3882 \expandafter\eqL@transpose@output@col
3883 \fi
3884 }

```

N.3 Measure

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

s@measure@line@begin

```

3885 \def\eqL@columns@measure@line@begin{%
3886 (dev)\eqL@dev{starting line \the\eqL@row@}%
3887 \global\eqL@column@\z@
3888 \global\eqL@line@height@\z@
3889 \global\eqL@line@depth@\z@
3890 \eqL@numbering@measure@line@begin
3891 \eqL@hook@linein
3892 }

3893 \def\eqL@columns@measure@cell{%
3894 \eqL@cellwidth@\wd\eqL@cellbox@
3895 \ifdefined\eqL@frame@cmd
3896 \eqL@frame@measure
3897 \advance\eqL@cellwidth@\eqL@frame@margin@
3898 \fi
3899 \ifdim\ht\eqL@cellbox@>\eqL@line@height@
3900 \global\eqL@line@height@\ht\eqL@cellbox@
3901 \fi
3902 \ifdim\dp\eqL@cellbox@>\eqL@line@depth@
3903 \global\eqL@line@depth@\dp\eqL@cellbox@
3904 \fi
3905 \ifnum\eqL@column@=\@ne
3906 \eqL@dimensions@startrow
3907 \fi
3908 \ifodd\eqL@column@
3909 \eqL@shape@pos@\tw@
3910 \else
3911 \eqL@shape@pos@\z@

```

```

3912 \fi
3913 \eq@shape@amount@z@
3914 \eq@dimensions@savecell
3915 \ifodd\eq@column@else
3916 \eq@dimensions@savesep
3917 \fi
3918 \kern\eq@cellwidth@
3919 }

```

ms@measure@line@end

```

3920 \def\eq@columns@measure@line@end{%
3921 \eq@punct@apply@line
3922 \eq@hook@lineout
3923 &\omit
3924 \ifnum\eq@column@>\eq@totalcolumns@
3925 \global\eq@totalcolumns@\eq@column@
3926 \fi

```

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

```

3927 \ifdefined\eq@frame@cmd
3928 \advance\eq@column@\@ne
3929 \wd\eq@cellbox@z@
3930 \eq@columns@measure@cell
3931 \fi
3932 \eq@measure@tag
3933 \eq@measure@endrow
3934 }

```

\eq@columns@measure

```

3935 \def\eq@columns@measure{%
3936 (dev)\eq@dev@enter\eq@columns@measure
3937 \eq@totalcolumns@z@
3938 \eq@measure@init\eq@columns@measure@line@begin\eq@columns@measure@line@end

3939 \setbox\z@\vbox{\measuring@true\halign{%
3940 &%
3941 \global\advance\eq@column@\@ne
3942 \global\let\eq@cell@container\@empty
3943 \global\setbox\eq@cellbox@hbox{%
3944 \eq@strut@cell
3945 \@lign
3946 $\m@th\displaystyle
3947 \eq@hook@colin
3948 ##%
3949 \eq@class@innerleft
3950 \eq@hook@innerleft
3951 $%
3952 }%
3953 \eq@cell@container
3954 \hfil
3955 \eq@columns@measure@cell
3956 \global\let\eq@frame@prevcmd\eq@frame@cmd
3957 &%
3958 \eq@prevwidth@\wd\eq@cellbox@

```

```

3959     \let\eql@frame@cmd\eql@frame@prevcmd
3960     \global\advance\eql@column@\@ne
3961     \global\let\eql@cell@container\@empty
3962     \setbox\eql@cellbox@\hbox{%
3963       \eql@strut@cell
3964       \@lign
3965       $\m@th\displaystyle
3966       \eql@hook@innerright
3967       \eql@class@innerright@sel
3968       ##%
3969       \eql@punct@apply@col
3970       \eql@hook@colout
3971       $%
3972     }%
3973     \eql@cell@container
3974     \eql@columns@measure@cell
3975     \hfil
3976     \crrr

3977     \noalign{%
3978       \eql@hook@blockbefore
3979     }%
3980     \eql@hook@blockin
3981     \eql@scan@body

3982     \ifvmode\else
3983       \eql@punct@apply@block
3984       \eql@hook@blockout
3985       \eql@display@endline
3986       \cr
3987     \fi
3988     \noalign{%
3989       \eql@hook@blockafter
3990     }%

```

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

```

3991     \omit
3992     \eql@column@\@ne
3993     \eql@columns@completerow
3994     \cr
3995   }}%

3996   \eql@measure@close

3997   \setbox\z@\vbox{%
3998     \unvbox\z@
3999     \unpenalty
4000     \global\setbox\@ne\lastbox
4001   }%
4002   \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4003   \let\eql@colwidth@tab\@empty
4004   \loop
4005     \setbox\@ne\hbox{%
4006       \unhbox\@ne
4007       \unskip
4008       \global\setbox\thr@@\lastbox

```

```

4009     }%
4010     \ifhbox\thr@@
4011         \eq@colwidth@save{\wd\thr@@}%
4012     \repeat

4013 <dev>\eq@dev@leave\eq@columns@measure
4014 }

```

N.4 Columns Placement

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

```

4015 \def\eq@columns@adjust{%
4016     \ifodd\eq@totalcolumns@
4017         \advance\eq@totalcolumns@ \@ne
4018     \fi
4019     \eq@tagpos@adjust@eval
4020     \eq@adjust@calc@columns
4021 }

```

N.5 Print

TODO: describe

ms@print@line@begin

```

4022 \def\eq@columns@print@line@begin{%
4023 <dev>\eq@dev{starting line \the\eq@row@}%
4024     \global\eq@column@\z@
4025     \global\eq@line@pos@\eq@marginleft@
4026     \global\eq@line@width@\z@
4027     \global\eq@line@avail@\eq@totalwidth@
4028     \global\eq@line@height@\z@
4029     \global\eq@line@depth@\z@
4030     \eq@numbering@print@line@begin
4031     \eq@hook@linein
4032 }

```

l@columns@print@cell

```

4033 \def\eq@columns@print@cell{%
4034     \eq@cellwidth@\wd\eq@cellbox@
4035     \ifodd\eq@column@
4036         \ifdefined\eq@frame@cmd
4037             \eq@frame@measure
4038             \advance\eq@cellwidth@\eq@frame@margin@
4039         \fi
4040         \dimen@\z@
4041     \else
4042         \advance\eq@cellwidth@-\eq@prevwidth@

```

draw a frame

```

4043     \ifdefined\eq@frame@cmd
4044         \eq@frame@measure
4045         \advance\eq@cellwidth@\eq@frame@margin@
4046         \advance\eq@prevwidth@\eq@frame@margin@

```

```

4047     \eql@frame@print
4048     \fi

```

update height and depth

```

4049     \ifdim\ht\eql@cellbox@>\eql@line@height@
4050         \global\eql@line@height@\ht\eql@cellbox@
4051     \fi
4052     \ifdim\dp\eql@cellbox@>\eql@line@depth@
4053         \global\eql@line@depth@\dp\eql@cellbox@
4054     \fi

```

print box

```

4055     \kern-\eql@prevwidth@
4056     \unhbox\eql@cellbox@
4057     \dimen@-\eql@cellwidth@
4058     \fi

```

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

```

4059     \advance\dimen@\eql@colwidth@get\eql@column@\relax
4060     \kern\dimen@

```

update available and used space

```

4061     \dimen@\eql@colwidth@get\eql@column@\relax
4062     \ifdim\eql@cellwidth@>\z@
4063         \ifdim\eql@line@width@=\z@
4064             \eql@line@avail@\eql@line@pos@
4065             \ifodd\eql@column@
4066                 \advance\eql@line@avail@\dimen@
4067                 \advance\eql@line@avail@-\eql@cellwidth@
4068             \fi
4069             \global\eql@line@avail@\eql@line@avail@
4070         \fi
4071         \eql@line@width@\eql@line@pos@
4072         \ifodd\eql@column@
4073             \advance\eql@line@width@\dimen@
4074         \else
4075             \advance\eql@line@width@\eql@cellwidth@
4076         \fi
4077         \global\eql@line@width@\eql@line@width@
4078     \fi
4079     \advance\eql@line@pos@\dimen@
4080     \ifodd\eql@column@\else
4081         \advance\eql@line@pos@\eql@colsep@
4082     \fi
4083     \global\eql@line@pos@\eql@line@pos@
4084 }

```

```

4085 \def\eql@columns@print@trailright{%
4086     &\omit
4087     \eql@prevwidth@\wd\eql@cellbox@
4088     \let\eql@frame@cmd\eql@frame@prevcmd
4089     \global\advance\eql@column@\@ne
4090     \eql@columns@print@cell
4091 }

```

lums@print@line@end

```

4092 \def\eql@columns@print@line@end{%
4093   \eql@punct@apply@line
4094   \eql@hook@lineout
4095 % \TODO add an even column with empty stuff if box processing deferred
4096   \ifodd\eql@column@
4097     \expandafter\eql@columns@print@trailright
4098   \fi
4099   \eql@columns@completerow
4100   \eql@columns@print@tag
4101 }

```

ql@columns@print@tag

```

4102 \def\eql@columns@print@tag{%
4103   \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4104   \eql@display@firstavail@set\eql@line@avail@
4105   \eql@columns@overfull
4106   \eql@numbering@print@line@eval
4107   \if@eqnsw
4108     \eql@tagbox@make\eql@composetag@print
4109   \fi
4110   \eql@tagpos@print@line@eval
4111   \eql@tagbox@print@cell
4112   \eql@tagpos@print@line@end
4113 }

```

\eql@columns@print

```

4114 \def\eql@columns@print{%
4115 (dev)\eql@dev@center\eql@columns@print
4116   \eql@shape@align@disable
4117   \eql@display@halign@init\eql@columns@print@line@begin
4118   \eql@display@halign@letcr\eql@columns@print@line@end
4119   \tabskip\eql@marginleft@

4120   \halign{%
4121     &%
4122     \global\advance\eql@column@ \@ne
4123     \global\let\eql@cell@container \@empty
4124     \global\setbox\eql@cellbox@ \hbox{%
4125       \eql@strut@cell
4126       \@lign
4127       $\m@th\displaystyle
4128       \eql@hook@colin
4129       ##%
4130       \eql@class@innerleft
4131       \eql@hook@innerleft
4132       \eql@tagging@mathsave
4133       $%
4134       \eql@tagging@mathaddlast
4135     }%
4136     \eql@cell@container
4137     \hfil
4138     \eql@columns@print@cell
4139     \global\let\eql@frame@prevcmd\eql@frame@cmd
4140     \tabskip\z@skip
4141     &%

```

```

4142 \eql@prevwidth@\wd\eql@cellbox@
4143 \let\eql@frame@cmd\eql@frame@prevcmd
4144 \global\advance\eql@column@\@ne
4145 \global\let\eql@cell@container\@empty
4146 \setbox\eql@cellbox@\hbox{%
4147   \unhbox\eql@cellbox@
4148   \eql@strut@cell
4149   \@lign
4150   $\m@th\displaystyle
4151   \eql@hook@innerright
4152   \eql@class@innerright@sel
4153   ##%
4154   \eql@punct@apply@col
4155   \eql@hook@colout
4156   \eql@tagging@mathsave
4157   $%
4158   \eql@tagging@mathaddlast
4159 }%
4160 \eql@cell@container
4161 \eql@columns@print@cell
4162 \hfil
4163 \tabskip\eql@colsep@\relax
4164 \crrr

4165 \noalign{%
4166   \eql@display@halign@start
4167   \eql@numbering@print@block@begin
4168   \eql@hook@blockbefore
4169 }%
4170 \eql@hook@blockin
4171 \eql@scan@body
4172 \ifvmode\else
4173   \relax
4174   \eql@punct@apply@block
4175   \eql@hook@blockout
4176   \eql@display@endline
4177   \cr
4178 \fi
4179 \noalign{%
4180   \eql@hook@blockafter
4181   \eql@display@halign@end
4182 (dev)\eql@dev@leave\eql@columns@print
4183 }%
4184 \eql@tagging@tablesalign
4185 }%
4186 }

```

O Interface

O.1 Scanning the Equation Body

The multi-line equatiuon 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.

```
4187 \newtoks\eql@scan@reg@
```

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

```
4188 \def\eql@scan@body@dump{\the\eql@scan@reg@}  
4189 \def\eql@scan@body@rescan{%  
4190   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}  
4191 \let\eql@scan@body\eql@scan@body@dump
```

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

```
4192 \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@`:

```
4193 \def\eql@scan@env#1{%  
4194   (dev)\eql@dev@enter\eql@scan@env  
4195   \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenvir}}%  
4196   \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`:

```
4197   \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%  
4198   \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate  
4199   \ifdefined\eql@scan@par  
4200     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate  
4201   \else  
4202     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar  
4203   \fi  
4204   \eql@scan@iterate  
4205 }
```

`\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:

```
4206 \long\def\eql@scan@env@iterate#1\end#2{%  
4207   \edef\eql@scan@stack{%
```

```

4208 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4209 \ifx\@empty\eql@scan@stack
4210 \@checkend{#2}%
4211 \eql@scan@addto{#1}%
4212 \expandafter\let\eql@scan@iterate\eql@scan@env@org
4213 (dev)\eql@dev@leave\eql@scan@env
4214 \expandafter\eql@scan@end
4215 \else
4216 \eql@scan@addto{#1\end{#2}}%
4217 \expandafter\eql@scan@iterate
4218 \fi
4219 }

```

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

```

4220 \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.

```

4221 \long\def\eql@scan@env@count#1\begin#2{%
4222 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4223 }

```

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

```

4224 \def\eql@scan@env@cancel{%
4225 \@namedef{end\@currentvir}{\ignorespacesafterend}%
4226 }

```

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 `\]`:

```

4227 \def\eql@scan@sqr#1{%
4228 (dev)\eql@dev@enter\eql@scan@sqr
4229 \def\eql@scan@end{#1\]}%
4230 \eql@scan@reg@\def\eql@scan@stack{b}%
4231 \let\eql@scan@sqr@org\[%\]
4232 \ifdefined\eql@scan@par
4233 \let\[\eql@scan@sqr@iterate%\]
4234 \else
4235 \let\[\eql@scan@sqr@iterate@nopar%\]
4236 \fi
4237 \[%\]
4238 }

```

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

```

4239 \long\def\eql@scan@sqr@iterate#1\]{%
4240 \edef\eql@scan@stack{%
4241 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4242 \ifx\@empty\eql@scan@stack
4243 \let\[\eql@scan@sqr@org%\]
4244 \eql@scan@addto{#1}%

```

```

4245 (dev)\eql@dev@leave\eql@scan@sqr
4246   \expandafter\eql@scan@end
4247   \else
4248     \eql@scan@addto{#1\}}%
4249     \expandafter\[%\]
4250   \fi
4251 }

```

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

```

4252 \def\eql@scan@sqr@iterate@nopar#1\{\eql@scan@sqr@iterate#1\}}

```

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

```

4253 \long\def\eql@scan@sqr@count#1\[#2{%\]
4254   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4255 }

```

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

```

4256 \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 `\>`:

```

4257 \def\eql@scan@ang#1{%
4258 (dev)\eql@dev@enter\eql@scan@ang
4259   \def\eql@scan@end{#1\>}%
4260   \eql@scan@reg@{\}\def\eql@scan@stack{b}%
4261   \let\eql@scan@ang@org\<%\>
4262   \ifdefined\eql@scan@par
4263     \let\<\eql@scan@ang@iterate%\>
4264   \else
4265     \let\<\eql@scan@ang@iterate@nopar%\>
4266   \fi
4267   \<%\>
4268 }

```

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

```

4269 \long\def\eql@scan@ang@iterate#1\>{%
4270   \edef\eql@scan@stack{%
4271     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4272   \ifx\@empty\eql@scan@stack
4273     \let\<\eql@scan@ang@org%\>
4274     \eql@scan@addto{#1}%
4275 (dev)\eql@dev@leave\eql@scan@ang
4276   \expandafter\eql@scan@end
4277   \else
4278     \eql@scan@addto{#1\>}%
4279     \expandafter\<%\>
4280   \fi
4281 }

```

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

```

4282 \def\eql@scan@ang@iterate@nopar#1\>\eql@scan@ang@iterate#1\>}

```

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

```
4283 \long\def\eql@scan@ang@count#1\<#2{%\>
4284   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4285 }
```

O.2 Options Processing

`\eqlequations@testall` The macro sequence started by `\eqlequations@testall` scans for optional arguments to the equation environments and appends them to the argument list using `\eqnaddopt`. All arguments are scanned such that any spaces stop the scanning and such that any alignment markers ‘&’ cannot interfere: **TODO:** update

```
4286 \def\eqlequations@testall{\eql@parseopt\eqlequations@parseopt}
4287 \def\eqlequations@parseopt{%
4288   \ifx\eql@parseopt@token*%
4289     \let\eql@parseopt@next\eql@parseopt@nonumber
4290   \fi
4291   \ifx\eql@parseopt@token!%
4292     \let\eql@parseopt@next\eql@parseopt@donumber
4293   \fi
4294   \ifx\eql@parseopt@token/%
4295     \let\eql@parseopt@next\eql@parseopt@transpose
4296   \fi
4297   \ifx\eql@parseopt@token[%]
4298     \let\eql@parseopt@next\eql@parseopt@opt
4299   \fi
4300   \ifx\eql@parseopt@token\eql@atxi
4301     \let\eql@parseopt@next\eql@parseopt@label
4302   \fi
4303   \ifx\eql@parseopt@token\eql@atxii
4304     \let\eql@parseopt@next\eql@parseopt@label
4305   \fi
4306   \ifx\eql@parseopt@token.%
4307     \let\eql@parseopt@next\eql@parseopt@punctdot
4308   \fi
4309   \ifx\eql@parseopt@token,%
4310     \let\eql@parseopt@next\eql@parseopt@punctcomma
4311   \fi
4312   \ifx\eql@parseopt@token~%
4313     \let\eql@parseopt@next\eql@parseopt@punctoff
4314   \fi
4315   \ifx\eql@parseopt@token-%
4316     \let\eql@parseopt@next\eql@parseopt@single
4317   \fi
4318   \ifx\eql@parseopt@token=%
4319     \let\eql@parseopt@next\eql@parseopt@lines
4320   \fi
4321   \ifx\eql@parseopt@token|%
4322     \let\eql@parseopt@next\eql@parseopt@columns
4323   \fi
4324   \ifx\eql@parseopt@token\label
4325     \let\eql@parseopt@next\eql@parseopt@end
4326   \fi
4327   \ifx\eql@parseopt@token\begin
4328     \let\eql@parseopt@next\eql@parseopt@end
4329   \fi
4330 }
```

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

```

4331 \def\eql@equations@processopt{%
4332   \let\eql@tags@container@block\eql@tags@container@clear
4333   \let\eql@tags@frame@cmd\@firstofone
4334   \let\eql@skip@force@above\@undefined
4335   \let\eql@skip@force@below\@undefined
4336   \let\eql@skip@force@leave\@undefined
4337   \let\eql@display@linewidth\@undefined
4338   \let\eql@display@marginleft\@undefined
4339   \let\eql@display@marginright\@undefined
4340   \eql@abovespace@\z@skip
4341   \eql@belowspace@\z@skip
4342   \eql@displaybreak@prepen@\@MM
4343   \eql@displaybreak@postpen@\@MM
4344   \eql@nextopt@process{equations}%
4345   \let\eql@punct@block\eql@punct@main
4346   \let\eql@punct@main\relax
4347   \eql@indent@\glueexpr\eql@indent@val\relax
4348   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4349 }

```

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:

```

4350 \def\eql@single@cr{%
4351   \eql@error{Cannot use '\string\\' within display equation.
4352     Please switch to equations environment}}%
4353 }

```

`\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:

```

4354 \def\eql@single@start{%
4355   \eql@display@enter
4356   \eql@tagging@start
4357   \eql@dollar@begin
4358   \eql@display@adjust
4359   \eql@numbering@init
4360   \eql@stack@save@equations
4361   \eql@numbering@single@init
4362   \ifdefined\eql@single@crerror\else
4363     \let\\ \eql@single@cr
4364   \fi
4365   \ifdefined\eql@single@native
4366     \let\eql@single@start@sel\eql@single@start@native
4367     \let\eql@single@end@sel\eql@single@end@native
4368   \else
4369     \let\eql@single@start@sel\eql@single@start@print
4370     \let\eql@single@end@sel\eql@single@end@print

```

```

4371 \fi
4372 \eql@single@start@sel
4373 }

```

`\eql@single@end` Closing code for single-line equation. Apply punctuation for the block, perform mode-specific ending, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```

4374 \def\eql@single@end{%
4375 \eql@punct@apply@block
4376 \eql@hook@eqout
4377 \eql@single@end@sel
4378 \eql@stack@restore
4379 \eql@dollar@dollar@end
4380 \eql@tagging@end
4381 \eql@display@leave
4382 }

```

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

```

4383 \def\eql@single@main{%
4384 \expandafter\eql@single@start
4385 \eql@scan@body
4386 \eql@single@end
4387 }

```

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

```

4388 \def\eql@mode@single{%
4389 \ifdefined\eql@single@doscan
4390 \let\eql@equations@main\eql@single@main
4391 \let\eql@equations@end\@empty
4392 \else
4393 \let\eql@equations@main\@undefined
4394 \let\eql@equations@end\eql@single@end
4395 \fi
4396 }

```

O.4 Multi-Line Main

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

```

4397 \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

```

4398 \def\eql@multi@main{%
4399 \eql@display@enter
4400 \eql@tagging@start
4401 \eql@dollar@dollar@begin
4402 \eql@display@adjust
4403 \eql@numbering@init
4404 \eql@stack@save@equations
4405 \ifdefined\eql@transpose@active

```

```

4406 \ifdefined\eq@multi@mode@lines\else
4407 \eq@transpose
4408 \fi
4409 \fi
4410 \ifdefined\eq@numbering@subeq@use
4411 \eq@numbering@subeq@init
4412 \fi
4413 \eq@display@init
4414 \let\intertext\eq@intertext
4415 \let\endintertext\endeq@intertext
4416 \eq@shape@align@enable

```

Now measure the given multi-line equations body:

```

4417 \ifdefined\eq@multi@mode@lines
4418 \eq@lines@measure
4419 \else
4420 \ifdefined\eq@ampproof@active
4421 \eq@ampproof
4422 \fi
4423 \eq@columns@measure
4424 \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 `\eq@lines@measure`:

```

4425 \ifdefined\eq@numbering@subeq@use
4426 \eq@numbering@subeq@test
4427 \fi
4428 \ifdefined\eq@multi@mode@lines\else
4429 \ifdefined\eq@multi@linesfallback
4430 \ifnum\eq@totalcolumns@=\@ne
4431 \let\eq@multi@mode@lines\eq@true
4432 \ifx\eq@multi@linesfallback\z@\else
4433 \eq@lines@measure
4434 \fi
4435 \fi
4436 \fi
4437 \fi

```

Adjust the multi-line equations body:

```

4438 \ifdefined\eq@multi@mode@lines
4439 \eq@lines@adjust
4440 \else
4441 \eq@columns@adjust
4442 \fi

```

Now print the multi-line equations body:

```

4443 \eq@display@print
4444 \eq@numbering@print@init
4445 \ifdefined\eq@multi@mode@lines
4446 \eq@lines@print
4447 \else
4448 \eq@columns@print
4449 \fi
4450 \eq@display@close

```

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

```

4451 \ifdefined\eq@numbering@subeq@use
4452   \eq@numbering@subeq@close
4453 \fi
4454 \eq@stack@restore
4455 \eq@dollar@dollar@end
4456 \eq@tagging@end
4457 \eq@display@leave
4458 }
```

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

```

\eq@mode@lines
4459 \def\eq@mode@columns{%
4460   \let\eq@equations@main\eq@multi@main
4461   \let\eq@equations@end\@empty
4462   \let\eq@multi@mode@lines\eq@false
4463 }
4464 \def\eq@mode@lines{%
4465   \let\eq@equations@main\eq@multi@main
4466   \let\eq@equations@end\@empty
4467   \let\eq@multi@mode@lines\eq@true
4468 }
```

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 `\eq@equations@start`:

```

4469 \newenvironment{equations}{%
4470 (dev)\eq@dev@enterenv
4471 \ifmmode
4472   \eq@error@mathmode{\string\begin{\@currenvir}}%
4473   \expandafter\eq@scan@env\expandafter\eq@scan@env@cancel
4474 \else
4475   \expandafter\eq@ampprotect\expandafter\eq@equations@testall
4476   \expandafter\eq@equations@start
4477 \fi
4478 }{%
4479 \eq@equations@end
4480 \ignorespacesafterend
4481 (dev)\eq@dev@leaveenv
4482 }
4483 \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`:

```

4484 \def\eq@equations@start{%
4485   \eq@equations@processopt
```



```

4486 \ifdefined\eqlequations@main
4487   \expandafter\eql@scan@env\expandafter\eqlequations@main
4488 \else
4489   \expandafter\eql@single@start
4490 \fi
4491 }

```

Square Brackets.

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

```

4492 \newenvironment{equations@sqr}{}{}
4493 \eql@markline@amsthm@register{equations@sqr}

```

`l@equations@sqr@open` Definition for ‘`\[`’. If already in math mode, ignore the enclosed contents. Otherwise add the default arguments `\eqlequations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eqlequations@sqr@start`:

```

4494 \protected\def\eqlequations@sqr@open{%
4495   \ifmmode
4496     \eql@error@mathmode{\string\[...\string\]}%
4497     \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
4498   \else
4499     (dev)\eql@dev@enter{\string\[...\string\]}%
4500     \expandafter\eqnaddopt\expandafter{\eqlequations@sqr@opt}%
4501     \begin{equations@sqr}%
4502     \let\]\eqlequations@sqr@close
4503     \expandafter\eql@ampprotect\expandafter\eqlequations@testall
4504     \expandafter\eqlequations@sqr@start
4505   \fi
4506 }

```

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

```

4507 \def\eqlequations@sqr@start{%
4508   \eqlequations@processopt
4509   \ifdefined\eqlequations@main
4510     \expandafter\eql@scan@sqr\expandafter\eqlequations@main
4511   \else
4512     \expandafter\eql@single@start
4513   \fi
4514 }

```

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

```

4515 \protected\def\eqlequations@sqr@close{%
4516   \eqlequations@end
4517   (dev)\eql@dev@leave{\string\]}%
4518   \end{equations@sqr}%
4519   \ignorespaces
4520 }

```

TODO: describe

```

\eql@sqr@open
\eql@sqr@close

```

```

4521 \let\eql@sqr@open\eql@equations@sqr@open
4522 \protected\def\eql@sqr@close{%
4523   \eql@error{'\string\}' may only close '\string\[']\}%\}
4524 }

```

Angle Brackets.

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

```

4525 \newenvironment{equations@ang}{}{}
4526 \newenvironment{equationsbox@ang}{}{}
4527 \eql@markline@amsthm@register{equations@ang}

```

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

```

4528 \protected\def\eql@ang@open{%
4529   (dev)\eql@dev@enter{\<...\string\>}%
4530   \ifmmode
4531     \expandafter\eqnadopt\expandafter{\eql@box@ang@opt}%
4532     \begin{equationsbox@ang}%
4533     \let\>\eql@box@ang@close
4534     \expandafter\eql@ampprotect\expandafter\eql@box@testall
4535     \expandafter\eql@box@start
4536   \else
4537     \expandafter\eqnadopt\expandafter{\eql@equations@ang@opt}%
4538     \begin{equations@ang}%
4539     \let\>\eql@equations@ang@close
4540     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4541     \expandafter\eql@equations@ang@start
4542   \fi
4543 }

```

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

```

4544 \protected\def\eql@ang@close{%
4545   \eql@error{'\string\>'} may only close '\string\<'\}%\>
4546 }

```

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

```

4547 \def\eql@equations@ang@start{%
4548   \eql@equations@processopt
4549   \ifdefined\eql@equations@main
4550     \expandafter\eql@scan@ang\expandafter\eql@equations@main
4551   \else
4552     \expandafter\eql@single@start
4553   \fi
4554 }

```

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

```

4555 \def\eql@equations@ang@close{%
4556   \eql@equations@end
4557   \end{equations@ang}%
4558   (dev)\eql@dev@leave{\<...\string\>}%
4559   \ignorespaces
4560 }

```

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

```

4561 \def\eql@box@ang@close{%
4562   \eql@box@end
4563   \end{equationsbox@ang}%
4564 (dev)\eql@dev@leave{\<...\string\>}%
4565   \ignorespaces
4566 }

```

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:

```

4567 \def\eql@decide@abovebelow#1#2#3#4#5{%
4568   \eql@decide@select{#1}{#2}{#3}{%
4569     {,abovebelow,both,tb}{#4#5},%
4570     {above,top,t}{#4},%
4571     {below,bottom,b}{#5}}}%

```

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

```

4572 \def\eql@decide@situation#1#2#3#4{%
4573   \eql@decide@select{#1}{#2}{#3}{%
4574     {{long}}{\def#4{0}}},%
4575     {{short}}{\def#4{1}}},%
4576     {{cont}}{\def#4{2}}},%
4577     {{par}}{\def#4{3}}},%
4578     {{top}}{\def#4{4}}},%
4579     {{noskip}}{\def#4{5}}},%
4580     {{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:

```

4581 \eql@define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{}%
4582   \eql@mode@stacked}
4583 \eql@define@key{equationsbox}{aligned,align,al,columns,col}[]{}%
4584   \eql@mode@aligned}
4585 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
4586 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
4587 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
4588 \eql@define@key{setup}{boxangopt}[]{}%
4589   \def\eql@box@ang@opt{columns,#1}}

```

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

```

4590 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@single}

```

```

4591 \eql@define@key{equations}{gathered,gather,ga,lines,ln}[]{%
4592   \eql@mode@lines}
4593 \eql@define@key{equations}{aligned,align,al,columns,col}[]{%
4594   \eql@mode@columns}
4595 \eql@define@key{equations,setup}{transpose}[true]{%
4596   \eql@decide@select{#3}{#2}{#1}{%
4597     {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
4598     {{noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
4599     {{\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}}%
4600 \eql@define@key{equations}{native}[true]{%
4601   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
4602   \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
4603 \eql@define@key{setup}{native}[true]{%
4604   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
4605 \eql@define@key{setup}{scanequation}[true]{%
4606   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
4607 \eql@define@key{setup}{sqropt}[]{%
4608   \def\eql@equations@sqr@opt{equation,#1}}
4609 \eql@define@key{setup}{angopt}[]{%
4610   \def\eql@equations@ang@opt{columns,#1}}

```

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

```

4611 \def\eql@keycat{equations,equationsbox,setup}
4612 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
4613 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
4614   {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
4615   {{cell}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
4616   {{tag}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
4617   {\eql@decide@true
4618     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
4619 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings concerning page breaks:

```

4620 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4621   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
4622   {{high,3}{\eql@displaybreak@pre3}},%
4623   {{med,medium,2}{\eql@displaybreak@pre2}},%
4624   {{low,1}{\eql@displaybreak@pre1}},%
4625   {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
4626   {{default,inherit,-1}{\eql@displaybreak@pre@mOne}}}}
4627 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4628   {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
4629   {{high,3}{\eql@displaybreak@post3}},%
4630   {{med,medium,2}{\eql@displaybreak@post2}},%
4631   {{low,1}{\eql@displaybreak@post1}},%
4632   {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
4633   {{default,inherit,-1}{\eql@displaybreak@post@mOne}}}}
4634 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4635   \eql@decide@select{#3}{#2}{#1}{%
4636     {{full,4}{\eql@displaybreak@inter4}},%
4637     {{high,3}{\eql@displaybreak@inter3}},%
4638     {{med,medium,2}{\eql@displaybreak@inter2}},%
4639     {{low,1}{\eql@displaybreak@inter1}},%
4640     {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
4641 \eql@define@key{equations}{prepenalty}{%
4642   \eql@displaybreak@prepen@numexpr#1\relax}

```

```

4643 \eqld@define@key{equations}{postpenalty}{%
4644   \eqld@displaybreak@postpen@numexpr#1\relax}
4645 \eqld@define@key{equations,setup}{interpenalty}{%
4646   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

4647 \eqld@define@key{control}{vspace}[]{\eqld@vspace@add{#1}}
4648 \eqld@define@key{control}{vspace*}[]{\eqld@vspace@addfixedbefore{#1}}
4649 \eqld@define@key{control}{vspace!}[]{\eqld@vspace@addfixedafter{#1}}
4650 \eqld@define@key{control}{break}[4]{\eqld@displaybreak@level[{#1}]}
4651 \eqld@define@key{control}{penalty}[]{\eqld@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4652 \eqld@define@key\eqld@keycat{displayheight}[strut]{%
4653   \eqld@decide@select{#3}{#2}{#1}{%
4654     {\eqld@decide@false{\let\eqld@display@height\@undefined}},%
4655     {\strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
4656     {\relax{\def\eqld@display@height{#1}}}}}%
4657 \eqld@define@key\eqld@keycat{displaydepth}[strut]{%
4658   \eqld@decide@select{#3}{#2}{#1}{%
4659     {\eqld@decide@false{\let\eqld@display@depth\@undefined}},%
4660     {\strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
4661     {\relax{\def\eqld@display@depth{#1}}}}}%

```

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

```

4662 \eqld@define@key{equations}{noskip}[]{%
4663   \eqld@decide@abovebelow{#3}{#2}{#1}%
4664   {\def\eqld@skip@force@above{5}}%
4665   {\def\eqld@skip@force@below{5}}}
4666 \eqld@define@key{equations}{short}[above]{%
4667   \eqld@decide@abovebelow{#3}{#2}{#1}%
4668   {\def\eqld@skip@force@above{1}}%
4669   {\def\eqld@skip@force@below{1}}}
4670 \eqld@define@key{equations}{long}[]{%
4671   \eqld@decide@abovebelow{#3}{#2}{#1}%
4672   {\def\eqld@skip@force@above{0}}%
4673   {\def\eqld@skip@force@below{0}}}
4674 \eqld@define@key{equations}{medskip}[]{%
4675   \eqld@decide@abovebelow{#3}{#2}{#1}%
4676   {\def\eqld@skip@force@above{6}}%
4677   {\def\eqld@skip@force@below{6}}}
4678 \eqld@define@key{equations}{par}[par]{%
4679   \eqld@decide@select{#3}{#2}{#1}{%
4680     {\default},{\let\eqld@skip@force@leave\@undefined}},%
4681     {\cont,hmode}{\let\eqld@skip@force@leave\z@}},%
4682     {\par,vmode}{\let\eqld@skip@force@leave\@ne
4683       \ifdefined\eqld@skip@force@below\else
4684         \def\eqld@skip@force@below{3}%
4685       \fi}},%
4686     {\top}{\let\eqld@skip@force@leave\tw@
4687       \ifdefined\eqld@skip@force@below\else
4688         \def\eqld@skip@force@below{4}
4689       \fi}}}%

```

Specify vertical spacing explicitly:

```

4690 \eqld@define@key{equations}{skip}{%

```

```

4691 \def\eql@skip@force@above{7}%
4692 \def\eql@skip@custom@above{#1}%
4693 \let\eql@skip@force@below\eql@skip@force@above
4694 \let\eql@skip@custom@below\eql@skip@custom@above}
4695 \eql@define@key{equations}{aboveskip}{%
4696 \def\eql@skip@force@above{7}%
4697 \def\eql@skip@custom@above{#1}}
4698 \eql@define@key{equations}{belowskip}{%
4699 \def\eql@skip@force@below{7}%
4700 \def\eql@skip@custom@below{#1}}
4701 \eql@define@key{equations}{abovespace}{%
4702 \advance\eql@abovespace@glueexpr#1\relax}
4703 \eql@define@key{equations}{belowspace}{%
4704 \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4705 \eql@define@key{intertext}{skip}{%
4706 \def\eql@skip@force@above{7}%
4707 \def\eql@skip@custom@above{#1}%
4708 \let\eql@skip@force@below\eql@skip@force@above
4709 \let\eql@skip@custom@below\eql@skip@custom@above}
4710 \eql@define@key{intertext}{aboveskip}{%
4711 \def\eql@skip@force@below{7}%
4712 \def\eql@skip@custom@below{#1}}
4713 \eql@define@key{intertext}{belowskip}{%
4714 \def\eql@skip@force@above{7}%
4715 \def\eql@skip@custom@above{#1}}
4716 \eql@define@key{intertext}{noskip}[]{%
4717 \eql@decide@abovebelow{#3}{#2}{#1}%
4718 {\def\eql@skip@force@below{5}}%
4719 {\def\eql@skip@force@above{5}}}
4720 \eql@define@key{intertext}{short}[]{%
4721 \eql@decide@abovebelow{#3}{#2}{#1}%
4722 {\def\eql@skip@force@below{1}}%
4723 {\def\eql@skip@force@above{1}}}
4724 \eql@define@key{intertext}{long}[]{%
4725 \eql@decide@abovebelow{#3}{#2}{#1}%
4726 {\def\eql@skip@force@below{0}}%
4727 {\def\eql@skip@force@above{0}}}
4728 \eql@define@key{intertext}{medskip}[]{%
4729 \eql@decide@abovebelow{#3}{#2}{#1}%
4730 {\def\eql@skip@force@below{6}}%
4731 {\def\eql@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

4732 \eql@define@key{setup}{skip,longskip}{%
4733 \abovedisplayskip\glueexpr#1\relax
4734 \belowdisplayskip\abovedisplayskip
4735 \def\eql@skip@long@above{#1}%
4736 \let\eql@skip@long@below\eql@skip@long@above}
4737 \eql@define@key{setup}{aboveskip,abovelongskip}{%
4738 \abovedisplayskip\glueexpr#1\relax
4739 \def\eql@skip@long@above{#1}}
4740 \eql@define@key{setup}{belowskip,belowlongskip}{%
4741 \belowdisplayskip\glueexpr#1\relax
4742 \def\eql@skip@long@below{#1}}
4743 \eql@define@key{setup}{aboveshortskip}{%
4744 \abovedisplayshortskip\glueexpr#1\relax

```

```

4745 \def\eql@skip@short@above{#1}}
4746 \eql@define@key{setup}{belowshortskip}{%
4747 \belowdisplayshortskip\glueexpr#1\relax
4748 \def\eql@skip@short@below{#1}}
4749 \eql@define@key{setup}{tagskip}{%
4750 \def\eql@skip@tag@above{#1}%
4751 \let\eql@skip@tag@below\eql@skip@tag@above}
4752 \eql@define@key{setup}{abovetagskip}{%
4753 \def\eql@skip@tag@above{#1}}
4754 \eql@define@key{setup}{belowtagskip}{%
4755 \def\eql@skip@tag@below{#1}}
4756 \eql@define@key{setup}{medskip}{%
4757 \def\eql@skip@med@above{#1}%
4758 \let\eql@skip@med@below\eql@skip@med@above}
4759 \eql@define@key{setup}{abovemedskip}{%
4760 \def\eql@skip@med@above{#1}}
4761 \eql@define@key{setup}{belowmedskip}{%
4762 \def\eql@skip@med@below{#1}}
4763 \eql@define@key{setup}{abovetopskip}{%
4764 \def\eql@skip@top@above{#1}}
4765 \eql@define@key{setup}{belowtopskip}{%
4766 \def\eql@skip@top@below{#1}}
4767 \eql@define@key{setup}{aboveparskip}{%
4768 \def\eql@skip@par@above{#1}}
4769 \eql@define@key{setup}{belowparskip}{%
4770 \def\eql@skip@par@below{#1}}
4771 \eql@define@key{setup}{abovecontskip}{%
4772 \eql@decide@select{#3}{#2}{#1}{%
4773 {hide}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%
4774 {relax{\def\eql@skip@cont@above{#1}}}}}%
4775 \eql@define@key{setup}{belowcontskip}{%
4776 \def\eql@skip@cont@below{#1}}
4777 \eql@define@key{setup}{shortmode}{%
4778 \eql@decide@select{#3}{#2}{#1}{%
4779 {{off,never,no}{\def\eql@skip@mode@short{0}}},%
4780 {{above,neverbelow,notbelow,belowoff}{\def\eql@skip@mode@short{1}}},%
4781 {{belowone,belowsingle}{\def\eql@skip@mode@short{2}}},%
4782 {{belowall,always,on}{\def\eql@skip@mode@short{3}}}}}%
4783 \eql@define@key{setup}{abovecontmode}{%
4784 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}
4785 \eql@define@key{setup}{belowcontmode}{%
4786 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
4787 \eql@define@key{setup}{aboveparmode}{%
4788 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
4789 \eql@define@key{setup}{belowparmode}{%
4790 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
4791 \eql@define@key{setup}{abovetopmode}{%
4792 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
4793 \eql@define@key{setup}{belowtopmode}{%
4794 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

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

```

4795 \def\eql@keycat{equations,subequations}
4796 \eql@define@key\eql@keycat{label}{\eql@tags@addblock@label{#1}}
4797 \eql@define@key\eql@keycat{labelname}{\eql@tags@addblock@name{#1}}
4798 \eql@define@key\eql@keycat{tag}{\eql@tags@addblock@tag{#1}}
4799 \eql@define@key\eql@keycat{tag*}{%

```

```

4800 \eql@tags@addblock@tagform@off\eql@tags@addblock@tag{#1}}
4801 \eql@define@key\eql@keycat{taglabel}{\eql@tags@addblock@ref{#1}}

```

TODO: describe

```

4802 \eql@define@key{control}{label}{\eql@tags@add@label{#1}}
4803 \eql@define@key{control}{labelname}{\eql@tags@add@name{#1}}
4804 \eql@define@key{control}{tag}{\eql@tags@add@tag{#1}}
4805 \eql@define@key{control}{tag*}{\eql@tags@add@tagform@off\eql@tags@add@tag{#1}}
4806 \eql@define@key{control}{taglabel}{\eql@tags@add@ref{#1}}
4807 \eql@define@key{control}{shiftright}{\eql@tags@add@raiseshift{#1}}
4808 \eql@define@key{control}{smashtag}{\eql@tags@add@raisesmash{#1}}
4809 \eql@define@key{control}{pushtag}{\eql@tags@add@forceraise}

```

TODO: describe

```

4810 \eql@define@key{setup}{labelname}{\protected@edef\eql@tags@name@generic{#1}}
4811 \eql@define@key{setup}{autolabel}[true]{%
4812   \eql@decide@bool{#3}{#2}{#1}\eql@tags@autolabel}
4813 \eql@define@key{setup}{autotag}[true]{%
4814   \eql@decide@bool{#3}{#2}{#1}\eql@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

4815 \def\eql@keycat{equations,setup}
4816 \eql@define@key\eql@keycat{tagmargin}[auto]{%
4817   \eql@decide@select{#3}{#2}{#1}{%
4818     {{auto,\eql@decide@false}{\let\eql@tagmargin@val\undefined}},%
4819     {\relax{\def\eql@tagmargin@val{#1}}}}}
4820 \eql@define@key\eql@keycat{tagmargin*}{%
4821   \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}}
4822 \eql@define@key\eql@keycat{tagmarginratio}{%
4823   \eql@tagmargin@ratio@dimexpr#1pt\relax}
4824 \eql@define@key\eql@keycat{tagmarginthreshold}{%
4825   \def\eql@tagmargin@threshold{#1}}
4826 \eql@define@key\eql@keycat{mintagsep}{\def\eql@tagsepmin@val{#1}}
4827 \eql@define@key\eql@keycat{mintagwidth}{%
4828   \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}}
4829 \eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin{#1}}
4830 \eql@define@key\eql@keycat{tagsnap}{%
4831   \eql@decide@select{#3}{#2}{#1}{%
4832     {\eql@decide@false{\let\eql@tagpos@snap\z@}},%
4833     {\relax{\def\eql@tagpos@snap{#1}}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

4834 \def\eql@keycat{equations,setup}
4835 \eql@define@key\eql@keycat{tagbox,taglayout}{%
4836   \eql@tags@taglayout@set{#1}}
4837 \eql@define@key\eql@keycat{tagbox*,taglayout*}{%
4838   \eql@tags@taglayout@set@direct{#1}}
4839 \eql@define@key\eql@keycat{tagform}{%
4840   \eql@tags@tagform@set{#1}}
4841 \eql@define@key\eql@keycat{tagform*}{%
4842   \eql@tags@tagform@set@direct{#1}}
4843 \eql@define@key\eql@keycat{subeqtemplate}{%
4844   \def\eql@subequations@template####1####2{#1}%
4845   \eql@append\eql@subequations@template{\theparentequation{equation}}}

```



```

4846 \eqld@define@key{control}{tagbox,taglayout}{%
4847   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
4848 \eqld@define@key{control}{tagbox*,taglayout*}{%
4849   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
4850 \eqld@define@key{control}{tagform}{%
4851   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
4852 \eqld@define@key{control}{tagform*}[{####1}]{%
4853   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

4854 \def\eqld@keycat{equations,setup}
4855 \eqld@define@key\eqld@keycat{numberline,number,num,numline,n}[all]{%
4856   \eqld@decide@select{#3}{#2}{#1}{%
4857     {{\eqld@decide@false,0,*}{\let\eqld@numbering@active\eqld@false}},%
4858     {{\eqld@decide@true,!}{\let\eqld@numbering@active\eqld@true}},%
4859     {{none,n,-}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
4860       \let\eqld@numbering@active\eqld@false}},%
4861     {{single,1}{\let\eqld@numbering@mode\eqld@numbering@mode@single
4862       \let\eqld@numbering@active\eqld@true}},%
4863     {{multi,@}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
4864       \let\eqld@numbering@active\eqld@true}},%
4865     {\relax{\eqld@numbering@set{#1}}}}}
4866 \eqld@define@key\eqld@keycat{nonumber,nn,*}[][%
4867   \let\eqld@numbering@active\eqld@false}
4868 \eqld@define@key\eqld@keycat{donumber,dn,!}[][%
4869   \let\eqld@numbering@active\eqld@true}
4870 \eqld@define@key\eqld@keycat{tagsleft,leqno}[]{\let\eqld@tagsleft\eqld@true}
4871 \eqld@define@key\eqld@keycat{tagsright,reqno}[]{\let\eqld@tagsleft\eqld@false}
4872 \eqld@define@key\eqld@keycat{tags,eqno}{%
4873   \eqld@decide@select{#3}{#2}{#1}{%
4874     {{right,r}{\let\eqld@tagsleft\eqld@false}},%
4875     {{left,l}{\let\eqld@tagsleft\eqld@true}}}}
4876 \eqld@define@key\eqld@keycat{evadetag,avoidtag}[true]{%
4877   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}
4878 \eqld@define@key\eqld@keycat{tagbetween}[true]{%
4879   \eqld@decide@bool{#3}{#2}{#1}\eqld@tagpos@doconvert}

```

TODO: describe

```

4880 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
4881 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
4882 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
4883 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

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

```

4884 \def\eqld@keycat{equations,setup}
4885 \eqld@define@key\eqld@keycat{layout}{\eqld@decide@select{#3}{#2}{#1}{%
4886   {{center,c}{\let\eqld@layoutleft\eqld@false}},%
4887   {{left,l}{\let\eqld@layoutleft\eqld@true}}}}
4888 \eqld@define@key\eqld@keycat{center}[]{\let\eqld@layoutleft\eqld@false}
4889 \eqld@define@key\eqld@keycat{flushleft,left}[]{\let\eqld@layoutleft\eqld@true}
4890 \eqld@define@key\eqld@keycat{leftmargin}{\def\eqld@layoutleftmargin{#1}}
4891 \eqld@define@key\eqld@keycat{leftmargin*}{%
4892   \settowidth\dimen{#1}\edef\eqld@layoutleftmargin{\the\dimen}}
4893 \eqld@define@key\eqld@keycat{minleftmargin}{%

```

```

4894 \def\eql@layoutleftmarginmin{#1}}
4895 \eql@define@key\eql@keycat{maxleftmargin}{%
4896 \eql@decide@select{#3}{#2}{#1}{%
4897 {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
4898 {\relax{\def\eql@layoutleftmarginmax{#1}}}}}

4899 \def\eql@keycat{equations,equationsbox}
4900 \eql@define@key\eql@keycat{margin}{%
4901 \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
4902 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}
4903 \eql@define@key\eql@keycat{marginright}{\def\eql@display@marginright{#1}}
4904 \eql@define@key\eql@keycat{equations}{linewidth,width}{\def\eql@display@linewidth{#1}}

```

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

```

4905 \def\eql@keycat{equations,setup}
4906 \eql@define@key\eql@keycat{alignshrink}{\eql@decide@select{#3}{#2}{#1}{%
4907 {\max,full,4}{\eql@alignbadness@inf@bad}},%
4908 {\high,3}{\eql@alignbadness@54\relax}},%
4909 {\med,medium,2}{\eql@alignbadness@18\relax}},%
4910 {\low,1}{\eql@alignbadness@6\relax}},%
4911 {\0,\eql@decide@false}{\eql@alignbadness@z@}}}}
4912 \eql@define@key\eql@keycat{tagshrink}{\eql@decide@select{#3}{#2}{#1}{%
4913 {\max,full,4}{\eql@tagbadness@inf@bad}},%
4914 {\high,3}{\eql@tagbadness@54\relax}},%
4915 {\med,medium,2}{\eql@tagbadness@18\relax}},%
4916 {\low,1}{\eql@tagbadness@6\relax}},%
4917 {\0,\eql@decide@false}{\eql@tagbadness@z@}}}}
4918 \eql@define@key\eql@keycat{alignbadness}{\eql@alignbadness@numexpr#1\relax}
4919 \eql@define@key\eql@keycat{tagbadness}{\eql@tagbadness@numexpr#1\relax}
4920 \eql@define@key\eql@keycat{mincolsep}{\eql@decide@select{#3}{#2}{#1}{%
4921 {\0,\eql@decide@false}{\def\eql@colsepmin@val{0pt}}},%
4922 {\relax{\def\eql@colsepmin@val{#1}}}}}
4923 \eql@define@key\eql@keycat{maxcolsep}{\eql@decide@select{#3}{#2}{#1}{%
4924 {\eql@decide@false{\def\eql@colsepmax@val{.5\maxdimen}}},%
4925 {\relax{\def\eql@colsepmax@val{#1}}}}}
4926 \eql@define@key\eql@keycat{fulllength}[true]{%
4927 \eql@decide@bool{#3}{#2}{#1}\eql@columns@fulllength}

4928 \eql@define@key\eql@keycat{equationsbox,setup}{colsep}{\eql@decide@select{#3}{#2}{#1}{%
4929 {\0,\eql@decide@false}{\def\eql@box@colsep{0pt}}},%
4930 {\relax{\def\eql@box@colsep{#1}}}}}
4931 \eql@define@key\eql@keycat{equations}{colsep}{\eql@decide@select{#3}{#2}{#1}{%
4932 {\0,\eql@decide@false}{\def\eql@box@colsep{0pt}}},%
4933 {\relax{\def\eql@box@colsep{#1}}}}}
4934 \let\eql@colsepmin@val\eql@box@colsep
4935 \let\eql@colsepmax@val\eql@box@colsep

```

Horizontal Shape. Configure horizontal alignment schemes:

```

4936 \def\eql@keycat{equations,equationsbox,setup}
4937 \eql@define@key\eql@keycat{shape}[default]{\eql@shape@set{#1}}
4938 \eql@define@key\eql@keycat{padding,pad}[indent]{%
4939 \eql@decide@select{#3}{#2}{#1}{%
4940 {\max}{\let\eql@paddingleft@val\undefined}},%
4941 {\indent}{\def\eql@paddingleft@val{\eql@indent@val}}},%
4942 {\0,\eql@decide@false}{\def\eql@paddingleft@val{0pt}}},%

```

```

4943   {\relax{\def\eql@paddingleft@val{#1}}}%
4944   \let\eql@paddingright@val\eql@paddingleft@val}
4945 \eql@define@key\eql@keycat{padleft}[indent]{%
4946   \eql@decide@select{#3}{#2}{#1}{%
4947     {\max}{\let\eql@paddingleft@val\@undefined}},%
4948     {\indent}{\def\eql@paddingleft@val{\eql@indent@val}}},%
4949     {\0,\eql@decide@false}{\def\eql@paddingleft@val{0pt}}},%
4950     {\relax{\def\eql@paddingleft@val{#1}}}}%
4951 \eql@define@key\eql@keycat{padright}[indent]{%
4952   \eql@decide@select{#3}{#2}{#1}{%
4953     {\max}{\let\eql@paddingright@val\@undefined}},%
4954     {\indent}{\def\eql@paddingright@val{\eql@indent@val}}},%
4955     {\0,\eql@decide@false}{\def\eql@paddingright@val{0pt}}},%
4956     {\relax{\def\eql@paddingright@val{#1}}}}%
4957 \eql@define@key\eql@keycat{indent}[2em]{%
4958   \def\eql@indent@val{#1}}

```

TODO: describe

```

4959 \eql@define@key{control}{align}[]{%
4960   \eql@decide@select{#3}{#2}{#1}{%
4961     {\l,left}{\global\eql@append\eql@cell@container{\eql@shape@pos@z}},%
4962     {\c,center}{\global\eql@append\eql@cell@container{\eql@shape@pos@one}},%
4963     {\r,right}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw}}}%
4964 \eql@define@key{control}{shift,shifto}[]{%
4965   \eql@decide@select{#3}{#2}{#1}{%
4966     {\*,indent}{\eql@shape@alignamount@set{\eql@indent}},%
4967     {\!,outdent}{\eql@shape@alignamount@set{-\eql@indent}},%
4968     {\relax{\eql@shape@alignamount@set{#1}}}}%
4969 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

4970 \def\eql@keycat{equations,equationsbox,setup}
4971 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
4972 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
4973 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
4974 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
4975 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
4976 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}{%
4977   {\ampeq,amprel,eafter,beforerel}\eql@class@ampeq},%
4978   {\eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}

```

Punctuation. Configure punctuation defaults:

```

4979 \def\eql@keycat{equations,equationsbox,setup}
4980 \eql@define@key\eql@keycat{punctsep}[,]{\def\eql@punct@sep{#1}}
4981 \eql@define@key\eql@keycat{punct}[.]{\def\eql@punct@main{#1}}
4982 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
4983 \eql@define@key\eql@keycat{punctline}[,]{\def\eql@punct@line{#1}}
4984 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
4985 \eql@define@key\eql@keycat{punctcol}[,]{\def\eql@punct@col{#1}}
4986 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}

4987 \eql@define@key{control}{punctsep}[,]{\def\eql@punct@sep{#1}}
4988 \eql@define@key{control}{punct}[.]{\def\eql@punct@block{#1}%
4989   \def\eql@punct@line{#1}\def\eql@punct@col{#1}}
4990 \eql@define@key{control}{punct*}[]{\let\eql@punct@block\relax}
4991 \eql@define@key{control}{punctapply}[]{\eql@punct@apply@block}

```

Frames. **TODO:** describe

```
4992 \eqld@define@key{equationsbox}{frame}[\fbox]{%
4993   \def\eql@box@frame{#1}%
4994   \ifx\eql@box@frame\empty\let\eql@box@frame\@firstofone\fi}
4995 \eqld@define@key{equationsbox}{wrap}[]{\eql@box@wrap#1}
```

TODO: describe

```
4996 \eqld@define@key{control}{framecell}[\fbox]{%
4997   \global\eqld@append\eqld@cell@container{\def\eqld@frame@cmd{#1}}
4998 \eqld@define@key{control}{frametag}[\fbox]{%
4999   \global\eqld@append\eqld@tags@container{\def\eqld@tags@frame@cmd{#1}}}
```

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

```
5000 \eqld@define@key{equations,equationsbox}{alt}{}
```

Injections.

```
5001 \eqld@define@key{control}{inject}{%
5002   \global\eqld@append\eqld@interline@container{%
5003     \eqld@append\eqld@display@injectbefore{#1}}}
5004 \eqld@define@key{control}{inject*}{%
5005   \global\eqld@append\eqld@interline@container{%
5006     \eqld@append\eqld@display@injectafter{#1}}}
5007 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
5008 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
5009 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
5010 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}
```

TODO: describe

```
5011 \eqld@define@key{markline}{pos}{%
5012   \eqld@decide@select{#3}{#2}{#1}{%
5013     {{below,push}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5014     {{baseline}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5015     {{bottom}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}
5016 \eqld@define@key{markline}{below,push}[]{%
5017   \let\eqld@markline@pos\eqld@markline@pos@below}
5018 \eqld@define@key{markline}{baseline}[]{%
5019   \let\eqld@markline@pos\eqld@markline@pos@baseline}
5020 \eqld@define@key{markline}{bottom}[]{%
5021   \let\eqld@markline@pos\eqld@markline@pos@bottom}
5022 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
5023 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}
5024 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
5025 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}
5026 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
5027 \eqld@define@key{setup}{markpos}{%
5028   \eqld@decide@select{#3}{#2}{#1}{%
5029     {{below}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5030     {{baseline}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5031     {{bottom}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}
```

Global Switches. Set global switches:

```
5032 \let\eqld@multi@linesfallback\eqld@false
```

```

5033 \let\eql@scan@par\eql@false
5034 \let\eql@single@crerror\eql@false
5035 \let\eql@ampproof@active\eql@false

5036 \eql@define@key{equations,setup}{linesfallback}[true]{%
5037   \eql@decide@select{#3}{#2}{#1}{%
5038     {\eql@decide@false{\let\eql@multi@linesfallback\eql@false}},%
5039     {{reuse,lean}{\let\eql@multi@linesfallback\z@}},%
5040     {{measure,full,\eql@decide@true}{\let\eql@multi@linesfallback\eql@true}}}}
5041 \eql@define@key{setup}{ampproof}[true]{%
5042   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
5043 \eql@define@key{setup}{crerror}[true]{%
5044   \eql@decide@bool{#3}{#2}{#1}\eql@single@crerror}
5045 \eql@define@key{setup}{modifierwarning}[true]{%
5046   \eql@decide@select{#3}{#2}{#1}{%
5047     {\eql@decide@false}{\let\eql@parseopt@warn\@empty}},%
5048     {\eql@decide@true}{\let\eql@parseopt@warn\eql@warn@parseopt}},%
5049     {{verbose,+}{\let\eql@parseopt@warn\eql@warn@parseopt@verbose}}}}
5050 \let\eql@parseopt@warn\eql@warn@parseopt
5051 \eql@define@key{equations,setup}{rescan}[true]{%
5052   \eql@decide@if{#3}{#2}{#1}%
5053     {\let\eql@scan@body\eql@scan@body@rescan}%
5054     {\let\eql@scan@body\eql@scan@body@dump}}
5055 \eql@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5056   \eql@decide@bool{#3}{#2}{#1}\eql@scan@par}
5057 \eql@define@key{setup}{defaults}{%
5058   \eql@decide@select{#3}{#2}{#1}{%
5059     {{classic}{\eql@defaults@classic}},%
5060     {{eqnlines}{\eql@defaults@eqnlines}}}}

```

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

```

5061 \let\eql@provide@opt@env\tw@
5062 \let\eql@provide@opt@amsmathends\eql@true
5063 \let\eql@provide@opt@backup\eql@false
5064 \let\eql@provide@opt@ang\eql@true
5065 \let\eql@provide@opt@eqref\eql@true

5066 \eql@define@key{setup}{amsmathends}[true]{%
5067   \eql@error@packageoption{#2}%
5068   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
5069 \eql@define@key{setup}{backup}[true]{%
5070   \eql@error@packageoption{#2}%
5071   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
5072 \eql@define@key{setup}{env}[equation]{%
5073   \eql@error@packageoption{#2}%
5074   \eql@decide@select{#3}{#2}{#1}{%
5075     {{none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5076     {{equation,latex}{\let\eql@provide@opt@env\@ne}},%
5077     {{amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}}
5078 \eql@define@key{setup}{ang}[true]{%
5079   \eql@error@packageoption{#2}%
5080   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5081 \eql@define@key{setup}{eqref}[true]{%
5082   \eql@error@packageoption{#2}%
5083   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Shortcut Options. **TODO:** describe

```

5084 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5085 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5086 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5087 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5088 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5089 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5090 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5091 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5092 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
5093 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
5094 \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

5095 \def\eql@defaults@classic{%
5096   \eqnlineset{numberline=all}%
5097   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5098   \eqnlineset{maxcolsep=off}%
5099   \eqnlineset{spread={\jot}}%
5100   \eqnlineset{tagmargin}%
5101   \eqnlineset{tagmarginratio=1}%
5102   \eqnlineset{tagmarginthreshold=0.5}%
5103   \eqnlineset{leftmargin={\leftmargini}}%
5104   \eqnlineset{padding=max}%
5105   \eqnlineset{evadetag=off}%
5106   \eqnlineset{displayheight=off}%
5107   \eqnlineset{displaydepth=off}%
5108   \eqnlineset{shortmode=belowsingle}%
5109   \eqnlineset{abovecontmode=short}%
5110   \eqnlineset{belowcontmode=short}%
5111   \eqnlineset{aboveparmode=long}%
5112   \eqnlineset{belowparmode=long}%
5113   \eqnlineset{abovetopmode=long}%
5114   \eqnlineset{belowtopmode=long}%
5115   \eqnlineset{abovelongskip={\abovedisplayskip}}%
5116   \eqnlineset{belowlongskip={\belowdisplayskip}}%
5117   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5118   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5119   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5120   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5121   \eqnlineset{abovecontskip=0pt}%
5122   \eqnlineset{belowcontskip=0pt}%
5123   \eqnlineset{aboveparskip=0pt}%
5124   \eqnlineset{belowparskip=0pt}%
5125   \eqnlineset{abovetopskip=0pt}%
5126   \eqnlineset{belowtopskip=0pt}%
5127   \eqnlineset{abovetagskip=0pt}%
5128   \eqnlineset{belowtagskip=0pt}%
5129   \eqnlineset{crrerror=false}%

```

```

5130 \eqnlineset{linesfallback=false}%
5131 }

```

values based on 10pt vs 12pt

`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):

```

5132 \def\eqn@defaults@eqnlines{%
5133   \eqnlineset{numberline=all}%
5134   \eqnlineset{mintagsep=.5em}%
5135   \eqnlineset{maxcolsep=2em}%
5136   \eqnlineset{spread={0.2\normalbaselineskip}}%
5137   \eqnlineset{tagmargin}%
5138   \eqnlineset{tagmarginratio=.334}%
5139   \eqnlineset{tagmarginthreshold=0.5}%
5140   \eqnlineset{leftmargin={\leftmargini}}%
5141   \eqnlineset{padding=0pt}%
5142   \eqnlineset{evadetag}%
5143   \eqnlineset{displayheight=strut}%
5144   \eqnlineset{displaydepth=strut}%
5145   \eqnlineset{shortmode=above}%
5146   \eqnlineset{abovecontmode=noskip}%
5147   \eqnlineset{belowcontmode=long}%
5148   \eqnlineset{aboveparmode=long}%
5149   \eqnlineset{belowparmode=long}%
5150   \eqnlineset{abovetopmode=noskip}%
5151   \eqnlineset{belowtopmode=long}%
5152   \eqnlineset{longskip={0.75\normalbaselineskip
5153     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5154   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5155     plus 0.25\normalbaselineskip}}%
5156   \eqnlineset{belowshortskip={0.0\normalbaselineskip
5157     plus 0.25\normalbaselineskip}}%
5158   \eqnlineset{medskip={0.4\normalbaselineskip
5159     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5160   \eqnlineset{abovecontskip=0pt}%
5161   \eqnlineset{belowcontskip=0pt}%
5162   \eqnlineset{aboveparskip=0pt}%
5163   \eqnlineset{belowparskip=0pt}%
5164   \eqnlineset{abovetopskip=0pt}%
5165   \eqnlineset{belowtopskip=0pt}%
5166   \eqnlineset{abovetagskip={0.25\normalbaselineskip
5167     minus 0.25\normalbaselineskip}}%
5168   \eqnlineset{belowtagskip={0.25\normalbaselineskip
5169     minus 0.25\normalbaselineskip}}%
5170   \eqnlineset{crerror=true}%
5171   \eqnlineset{linesfallback=true}%
5172 }

```

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 5173 \def\eql@provide@movecmd#1#2{%
\eql@provide@movestar 5174   \eql@letcs{#1\expandafter}\csname #2\endcsname
@provide@undefinecmd 5175 }
@provide@undefineenv 5176 \def\eql@provide@moveenv#1#2{%
                    5177   \eql@provide@movecmd{#1}{#2}%
                    5178   \eql@markline@amsthm@register{#1}%
                    5179   \ifcsname end#2\endcsname
                    5180     \eql@provide@movecmd{end#1}{end#2}%
                    5181   \fi
                    5182 }
                    5183 \def\eql@provide@movestar#1#2{%
                    5184   \eql@provide@moveenv{#1}{#2}%
                    5185   \ifcsname #2*\endcsname
                    5186     \eql@provide@moveenv{#1*}{#2*}%
                    5187   \fi
                    5188 }
                    5189 \def\eql@provide@undefinecmd#1{%
                    5190   \eql@letcs{#1}\@undefined
                    5191 }
                    5192 \def\eql@provide@undefineenv#1{%
                    5193   \eql@provide@undefinecmd{#1}%
                    5194   \eql@provide@undefinecmd{end#1}%
                    5195 }
```

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:

```
5196 \def\eql@amsmath@endfix#1#2{%
5197   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
5198   \expandafter\ifx\csname end#1\endcsname\@tempa
5199     \eql@provide@movecmd{end#1}{end#2}%
5200   \fi
5201 }
```

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

```
5202 \def\eql@amsmath@fixends{%
5203   \eql@amsmath@after{%
5204     \eql@amsmath@endfix{gather*}{gather}%
5205     \eql@amsmath@endfix{multline*}{multline}%
5206     \eql@amsmath@endfix{align*}{align}%
5207     \eql@amsmath@endfix{flalign*}{align}%
5208     \eql@amsmath@endfix{flalign*}{align}%
5209     \eql@amsmath@endfix{alignat*}{align}%
5210     \eql@amsmath@endfix{alignat*}{align}%
5211     \eql@amsmath@endfix{xalignat*}{align}%
5212   }
```



```

5212 \eql@amsmath@endfix{xalignat*}{align}%
5213 \eql@amsmath@endfix{xxalignat}{align}%
5214 \eql@amsmath@endfix{gathered}{aligned}%
5215 \eql@amsmath@endfix{alignedat}{aligned}%
5216 }
5217 }

```

Backup amsmath Environments. We can backup all amsmath environments *env* to *amsenv* so that they can be used in parallel if needed.

provide@backup@amsmath Copy an amsmath environment *env* to *amsenv* whenever amsmath is loaded: **TODO:** amsthm

```

5218 \def\eql@provide@backup@amsmath#1{%
5219 \eql@amsmath@after{%
5220 \eql@provide@moveenv{ams#1}{#1}%
5221 \AddToHook{package/amsthm/after}{\eql@provide@movecmd{ams#1@qed}{#1@qed}}%
5222 }%
5223 }

```

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

```

5224 \def\eql@provide@backup@eqref{%
5225 \eql@amsmath@after{%
5226 \eql@provide@movecmd{amseqref}{eqref}%
5227 }%
5228 }

```

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

```

5229 \def\eql@provide@backup@multlined{%
5230 \AddToHook{package/mathtools/after}{%
5231 \eql@provide@moveenv{amsmultlined}{multlined}%
5232 }%
5233 }

```

provide@backup@equation The L^AT_EX 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:

```

5234 \def\eql@provide@backup@equation{%
5235 \eql@amsmath@after{%
5236 \eql@provide@moveenv{amsequation}{equation}%
5237 \eql@tagging@register@env{amsequation}%
5238 \eql@provide@moveenv{amsequation*}{equation*}%
5239 \eql@tagging@register@env{amsequation*}%
5240 \AddToHook{package/amsthm/after}{%
5241 \eql@provide@movecmd{amsequation*@qed}{equation*@qed}}%
5242 }%
5243 \AddToHook{package/hyperref/after}{%
5244 \ifpackageloaded{amsmath}{}%

```

```

5245 \let\latexequation\H@equation
5246 \let\endlatexequation\H@endequation
5247 \eqL@tagging@register@env{latexequation}%
5248 \eqL@provide@moveenv{hyperrefequation}{equation}%
5249 \eqL@tagging@register@env{hyperrefequation}%
5250 \AddToHook{package/amsthm/after}{%
5251   \eqL@provide@movecmd{latexequation@qed}{equation@qed}%
5252   \eqL@provide@movecmd{hyperequation@qed}{equation@qed}
5253 }%
5254 }%
5255 }%
5256 \@ifpackageloaded{amsmath}{\@ifpackageloaded{hyperref}}{}{%
5257   \eqL@provide@moveenv{latexequation}{equation}%
5258   \eqL@tagging@register@env{latexequation}%
5259   \AddToHook{package/amsthm/after}{%
5260     \eqL@provide@movecmd{latexequation@qed}{equation@qed}}%
5261   }%
5262 }

```

`\eqL@backup@displaymath` **TODO:** describe

```

5263 \def\eqL@provide@backup@displaymath{%
5264   \eqL@provide@moveenv{latexdisplaymath}{displaymath}%
5265   \AddToHook{package/amsthm/after}{%
5266     \eqL@provide@movecmd{latexdisplaymath@qed}{displaymath@qed}}%
5267 }

```

`\eqL@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:

```

5268 \def\eqL@provide@backup@subequations{%
5269   \eqL@amsmath@after{%
5270     \eqL@provide@moveenv{amssubequations}{subequations}%
5271   }%
5272   \AddToHook{package/hyperref/after}{%
5273     \AddToHook{cmd/amssubequations/before}{%
5274       {%
5275         \stepcounter{equation}%
5276         \protected@edef\theHparentequation{\theHequation}%
5277         \addtocounter{equation}{-1}%
5278       }%
5279     \AddToHook{cmd/amssubequations/after}{%
5280       {%
5281         \def\theHequation{\theHparentequation\alph{equation}}%
5282         \ignorespaces
5283       }%
5284     }%
5285 }

```

`\eqL@provide@backup` Backup all `amsmath` environments:

```

5286 \def\eqL@provide@backup{%
5287   \eqL@provide@backup@eqref
5288   \eqL@provide@backup@equation
5289   \eqL@provide@backup@displaymath
5290   \eqL@provide@backup@amsmath{gather}%
5291   \eqL@provide@backup@amsmath{gather*}%
5292   \eqL@provide@backup@amsmath{multline}%

```

```

5293 \eqL@provide@backup@amsmath{multline*}%
5294 \eqL@provide@backup@amsmath{align}%
5295 \eqL@provide@backup@amsmath{align*}%
5296 \eqL@provide@backup@amsmath{flalign}%
5297 \eqL@provide@backup@amsmath{flalign*}%
5298 \eqL@provide@backup@amsmath{alignat}%
5299 \eqL@provide@backup@amsmath{alignat*}%
5300 \eqL@provide@backup@amsmath{xalignat}%
5301 \eqL@provide@backup@amsmath{xalignat*}%
5302 \eqL@provide@backup@amsmath{xxalignat}%
5303 \eqL@provide@backup@amsmath{aligned}%
5304 \eqL@provide@backup@amsmath{aligned*}%
5305 \eqL@provide@backup@amsmath{alignedat}%
5306 \eqL@provide@backup@amsmath{alignedat*}%
5307 \eqL@provide@backup@amsmath{gathered}%
5308 \eqL@provide@backup@amsmath{gathered*}%
5309 \eqL@provide@backup@multlined
5310 \eqL@provide@backup@subequations
5311 }

```

Replacement amsmath Environments. **TODO:** describe

```

5312 \def\eqL@alignat@gobblecol#1{%
5313   \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.)
5314 \newenvironment{eqL@gathered}
5315   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5316 \newenvironment{eqL@multlined}
5317   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5318 \newenvironment{eqL@aligned}
5319   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
5320 \newenvironment{eqL@alignedat}
5321   {\eqnaddopt{columns,colsep=off}\eqL@alignat@gobblecol\equationsbox}
5322   {\endequationsbox}

```

`eqL@equation` (*env.*) Define replacement versions for display environments `equation`, `gather`, `multline`,
`eqL@gather` (*env.*) `aligned` and derivatives which forward to `equations` with specific presets: **TODO:**
`eqL@multline` (*env.*) `amsmath` at variants would need predefined columns for full operation

```

eqL@align (env.)
5323 \newenvironment{eqL@equation}
5324   {\eqnaddopt{equation}\equations}{\endequations}
5325 \newenvironment{eqL@displaymath}
5326   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
5327 \newenvironment{eqL@gather}
5328   {\eqnaddopt{lines}\equations}{\endequations}
5329 \newenvironment{eqL@multline}
5330   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5331   {\endequations}
5332 \newenvironment{eqL@align}
5333   {\eqnaddopt{columns}\equations}{\endequations}
5334 \newenvironment{eqL@flalign}
5335   {\eqnaddopt{fulllength}\eqL@align}{\endequations}
5336 \newenvironment{eqL@alignat}
5337   {\eqnaddopt{colsep=off}\eqL@xalignat}{\endequations}
5338 \newenvironment{eqL@xalignat}

```

```

5339 {\eql@alignat@gobblecol\eql@align}{\endequations}
5340 \newenvironment{eql@xxalignat}
5341 {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
5342 \newenvironment{eql@equation*}
5343 {\eqnaddopt{nonumber}\eql@equation}{\endequations}
5344 \newenvironment{eql@gather*}
5345 {\eqnaddopt{nonumber}\eql@gather}{\endequations}
5346 \newenvironment{eql@multline*}
5347 {\eqnaddopt{nonumber}\eql@multline}{\endequations}
5348 \newenvironment{eql@align*}
5349 {\eqnaddopt{nonumber}\eql@align}{\endequations}
5350 \newenvironment{eql@flalign*}
5351 {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
5352 \newenvironment{eql@alignat*}
5353 {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
5354 \newenvironment{eql@xalignat*}
5355 {\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):

```

5356 \def\eql@provide@onlyonce#1#2{%
5357   \def\eql@tmp{#2}%
5358   \def\@tempa{#1}%
5359   \ifx\eql@tmp\@tempa
5360     \let\eql@tmp\@empty
5361   \fi
5362   \ifx\eql@tmp\@empty
5363     \let\eql@tmp\@undefined
5364     \ifx\@nodocument\relax
5365       \def\eql@tmp{#1}%
5366     \fi
5367     \ifcsname eql@provided@#1\endcsname
5368       \def\eql@tmp{#1}%
5369     \fi
5370     \eql@letcs{eql@provided@#1}\eql@true
5371   \else
5372     \def\@tempa{*}%
5373     \ifx\eql@tmp\@tempa
5374       \def\eql@tmp{#1}%
5375     \fi
5376   \fi
5377 }

```

\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:

```

5378 \def\eql@provide@eqref#1{%
5379   \def\eql@tmp{#1}%

```

```

5380 \def\@tempa{\eqref}%
5381 \ifx\eq\@tmp\@tempa
5382   \let\eq\@tmp\@empty
5383 \fi
5384 \ifx\eq\@tmp\@empty
5385   \eq\@provide@onlyonce{eqref}{}%
5386 \else
5387   \def\@tempa{*}%
5388   \ifx\eq\@tmp\@tempa
5389     \def\eq\@tmp{eqref}%
5390   \else
5391     \edef\eq\@tmp{\expandafter\@gobble\string#1}%
5392   \fi
5393 \fi
5394 \ifdefined\eq\@tmp
5395   \expandafter\eq\@provide@movecmd\expandafter{\eq\@tmp}{eq\@eqref}%
5396 \else
5397   \eq\amsmath@after{%
5398     \eq\@provide@movecmd{eqref}{eq\@eqref}%
5399   }%
5400   \AddToHook{package/mathtools/after}{%
5401     \eq\@provide@movecmd{eqref}{eq\@eqref}%
5402   }%
5403   \eq\@provide@movecmd{eqref}{eq\@eqref}%
5404   \eq\amsmath@undefine\eqref
5405 \fi
5406 }

```

```

5407 \def\eqL@provide@amsmath#1#2{%
5408   \eqL@provide@onlyonce{#1}{#2}%
5409   \ifdefined\eqL@tmp
5410     \expandafter\eqL@provide@movestart\expandafter{\eqL@tmp}{eqL@#1}%
5411   \else
5412     \eqL@amsmath@after{%
5413       \eqL@provide@movestart{#1}{eqL@#1}%
5414     }%
5415     \AddToHook{package/mathtools/after}{%
5416       \eqL@provide@movestart{#1}{eqL@#1}%
5417     }%
5418     \eqL@provide@movestart{#1}{eqL@#1}%
5419     \eqL@amsmath@before{\eqL@provide@undefineenv{#1}}%
5420     \ifcsname eqL@#1*\endcsname
5421       \eqL@amsmath@before{\eqL@provide@undefineenv{#1*}}%
5422     \fi
5423   \fi
5424 }

```

```

5425 \def\eq@provide@multlined#1{%
5426   \eq@provide@onlyonce{multlined}{#1}%
5427   \ifdefined\eq@tmp
5428     \expandafter\eq@provide@moveenv\expandafter{\eq@tmp}{\eq@multlined}%

```

```

5429 \else
5430   \AddToHook{package/mathtools/after}{%
5431     \eql@provide@moveenv{multlined}{eql@multlined}%
5432   }%
5433   \eql@provide@moveenv{multlined}{eql@multlined}%
5434   \ifpackage{mathtools}{\AddToHook{package/mathtools/before}{%
5435     \eql@provide@undefineenv{multlined}}}%
5436 \fi
5437 }

```

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:

```

5438 \def\eql@provide@equation#1{%
5439   \eql@provide@onlyonce{equation}{#1}%
5440   \ifdefined\eql@tmp
5441     \expandafter\eql@provide@movestar\expandafter{\eql@tmp}{eql@equation}%
5442   \else
5443     \eql@amsmath@after{%
5444       \eql@provide@movestar{equation}{eql@equation}%
5445     }%
5446     \AddToHook{package/hyperref/after}{%
5447       \ifpackage{amsmath}{}%
5448       \eql@provide@moveenv{equation}{eql@equation}%
5449     }%
5450   }%
5451   \eql@provide@movestar{equation}{eql@equation}%
5452   \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5453   \ifdefined\eql@tagging@on
5454     \AddToHook{begindocument/end}{%
5455       \eql@provide@movestar{equation}{eql@equation}%
5456     }%
5457 \fi
5458 \fi
5459 }

```

@provide@displaymath **TODO:** describe

```

5460 \def\eql@provide@displaymath#1{%
5461   \eql@provide@onlyonce{displaymath}{#1}%
5462   \ifdefined\eql@tmp
5463     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@displaymath}%
5464   \else
5465     \eql@provide@moveenv{displaymath}{eql@displaymath}%
5466     \ifdefined\eql@tagging@on
5467       \AddToHook{begindocument/end}{%
5468         \eql@provide@moveenv{displaymath}{eql@displaymath}%
5469       }%
5470   \fi
5471 \fi
5472 }

```

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:

```

5473 \def\eql@provide@subequations#1{%
5474   \eql@provide@onlyonce{subequations}{#1}%
5475   \ifdefined\eql@tmp
5476     \expandafter\eql@provide@moveenv
5477     \expandafter{\eql@tmp}{eql@subequations}%
5478   \else
5479     \eql@amsmath@after{%
5480       \eql@provide@moveenv{subequations}{eql@subequations}%
5481     }%
5482     \AddToHook{package/hyperref/after}{%
5483       \AddToHook{cmd/subequations/before}[hyperref]{}%
5484       \AddToHook{cmd/subequations/after}[hyperref]{}%
5485       \RemoveFromHook{cmd/subequations/before}[hyperref]%
5486       \RemoveFromHook{cmd/subequations/after}[hyperref]%
5487       \AddToHook{begindocument/end}{%
5488         \eql@provide@moveenv{subequations}{eql@subequations}%
5489       }%
5490     }%
5491     \eql@provide@moveenv{subequations}{eql@subequations}%
5492     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}}%
5493 \fi
5494 }
```

`\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:

```

5495 \def\eql@provide@sqr{%
5496   \let\[ \eql@sqr@open
5497   \let] \eql@sqr@close
5498   \eql@amsmath@after{%
5499     \let\[ \eql@sqr@open
5500     \let] \eql@sqr@close
5501   }%
5502   \ifdefined\eql@tagging@on
5503     \AddToHook{begindocument/end}{%
5504       \let\[ \eql@sqr@open
5505       \let] \eql@sqr@close
5506     }%
5507 \fi
5508 }
```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```

5509 \def\eql@provide@ang{%
5510   \let\< \eql@ang@open
5511   \let\> \eql@ang@close
5512 }
```

Interface.

`provide` (*key*) We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

5513 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5514 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5515 \eql@define@key{provide}{gather}[]{\eql@provide@amsmath{gather}{#1}}
5516 \eql@define@key{provide}{multline}[]{\eql@provide@amsmath{multline}{#1}}
5517 \eql@define@key{provide}{align}[]{\eql@provide@amsmath{align}{#1}}
5518 \eql@define@key{provide}{flalign}[]{\eql@provide@amsmath{flalign}{#1}}
5519 \eql@define@key{provide}{alignat}[]{\eql@provide@amsmath{alignat}{#1}}
5520 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsmath{xalignat}{#1}}
5521 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsmath{xxalignat}{#1}}
5522 \eql@define@key{provide}{aligned}[]{\eql@provide@amsmath{aligned}{#1}}
5523 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsmath{alignedat}{#1}}
5524 \eql@define@key{provide}{gathered}[]{\eql@provide@amsmath{gathered}{#1}}
5525 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5526 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5527 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
5528 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
5529 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
5530 \eql@define@key{provide}{tagform}[]{%
5531   \def\tagform@##1{\maketag@@@{\eql@tags@tagform{#1}}}}
5532 \eql@define@key{provide}{maketag}[]{%
5533   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}

```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```

5534 \newcommand{\eqnlinesprovide}[1]{%
5535 (dev)\eql@dev@start\eqnlinesprovide
5536   \eql@setkeys{provide}{#1}%
5537   \ignorespaces
5538 }

```

P.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
5539 \let\eql@error@packageoption\gobble
```

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

```

5540 \DeclareOption{leqno}{\eqnlinesset{tagsleft}}
5541 \DeclareOption{fleqn}{\eqnlinesset{left}}

```

Pass undeclared options on to keyval processing:

```
5542 \DeclareOption*{\expandafter\eqnlinesset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

5543 \eql@defaults@eqnlines
5544 \eql@mode@columns
5545 \eql@mode@aligned

```

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

```

5546 \ifdefined\tagsleft@true\else
5547   \expandafter\newif\csname iftagsleft@\endcsname
5548 \fi
5549 \ifdefined\@fleqntrue\else
5550   \expandafter\newif\csname if@fleqn\endcsname

```



```
5551 \fi
```

Import amsmath switches leqno as tagsleft and fleqn as left:

```
5552 \eq@amsmath@after{%
5553   \ifnum\eq@provide@opt@env=\tw@
5554     \iftagsleft@
5555       \eqnlineset{tags=left}%
5556     \else
5557       \eqnlineset{tags=right}%
5558     \fi
5559     \if@fleqn
5560       \eqnlineset{layout=left}%
5561     \else
5562       \eqnlineset{layout=center}%
5563     \fi
5564 \fi
5565 }
```

Process package options:

```
5566 \ProcessOptions
```

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

```
5567 \def\eq@error@packageoption#1{%
5568   \eq@error{may only use '1' as a package option}%
5569 }
```

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

```
5570 \ifdefined\eq@provide@opt@amsmathends\eq@amsmath@fixends\fi
```

Backup all amsmath environments that may be overwritten to `ams...`. This will happen before any replacements:

```
5571 \ifdefined\eq@provide@opt@backup\eq@provide@backup\fi
```

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

```
5572 \ifnum\eq@provide@opt@env>\z@
5573   \eqnlinesprovide{equation,sqr,displaymath}
5574 \fi
```

Provide amsmath equation environments if desired:

```
5575 \ifnum\eq@provide@opt@env=\tw@
5576   \eqnlinesprovide{%
5577     multline,gather,align,flalign,alignat,xalignat,xxalignat,%
5578     multlined,gathered,aligned,alignedat,%
5579     subequations}
5580 \fi
```

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

```
5581 \ifdefined\eq@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
5582 \ifdefined\eq@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```