

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`

2025/03/11, v0.6

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

– preview version –

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	6
2.3	Horizontal Adjustment	7
2.4	Punctuation	8
2.5	Math Classes at Alignment	9
2.6	Vertical Spacing	9
2.7	Further Environments	11
2.8	General Options	11
2.9	Feature Selection and Package Options	12
3	Information	12
3.1	Copyright	12
3.2	Credits	12
3.3	Files and Installation	13
3.4	Related CTAN Packages	13
3.5	Feature Suggestions	14
3.6	Revision History	14
A	Implementation	14

B	General Support	15
	B.1 Development Messages	15
	B.2 Look-Ahead in Alignment	15
	B.3 Error Messages	17
	B.4 amsmath Integration	17
C	Parameters and Registers	18
	C.1 Supporting Definitions	18
	C.2 Parameters	18
	C.3 Registers	21
	C.4 Hooks	22
D	Tagging Support	23
E	Punctuation	24
F	Math Classes at Alignment	25
G	Equations Box Environment	26
H	Equation Numbering	30
	H.1 Tag Formatting	30
	H.2 Showkeys Integration	31
	H.3 Labels	32
	H.4 Tags	32
	H.5 Anchors	33
	H.6 Tag Composition	33
	H.7 Tagbox Methods	35
	H.8 Numbering Schemes	36
	H.9 Numbering Framework	39
I	Subequation Numbering	43
	I.1 Definitions	43
	I.2 Environment	44
	I.3 Subequation Scheme	45
J	Display Equations Support	46
	J.1 Display Breaks	46
	J.2 General Initialisation	47
	J.3 halign Support	50
	J.4 Stack	52
K	Horizontal Spacing for Lines	53
	K.1 Supporting Definitionss	53
	K.2 Shape Schemes	54
	K.3 Adjustment Methods	57
	K.4 Centered Layout	58
	K.5 Flush-Left Layout	63
L	Single-Line Equation	66
	L.1 Environment	66
	L.2 Native	67
	L.3 Adjustment	68
M	Multi-Line Support	68

M.1	Registers	68
M.2	Measure Support	69
M.3	Print Support	70
M.4	Line Breaks	71
M.5	Intertext	71
M.6	Main	74
N	Multi-Line Lines Mode	75
N.1	Measure	75
N.2	Print	77
O	Multi-Line Align Mode	78
O.1	Registers	79
O.2	General Processing, Preamble	79
O.3	Print	79
O.4	Measure	83
O.5	Adjust	85
P	Interface	90
P.1	Scanning the Environment's Body	90
P.2	Options Processing	92
P.3	Environments	93
P.4	Independent Routines of <code>amsmath</code>	95
Q	Options	95
Q.1	Selection Tools	96
Q.2	Declaration Code	97
Q.3	Parameter Sets	97
Q.4	Options Declarations	99
Q.5	Component Selection	105
Q.6	Global and Package Options	110

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-lined equations with alignment. Some of its deficiencies were addressed by the multi-line equation environments of the package `amsmath` which have become an established standard for these purposes.

The package `eqnlines` builds upon and extends the functionality of the $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and `amsmath` equation environments with some new features as well as convenient options to adjust the layout where needed. The main additions are as follows:

- Equation numbers can be assigned to individual lines (as for `align` and `gather`) or once for the multi-line equation block (as for `multline`). In the former case, a sub-numbering scheme can be applied (as through `subequations`). In the latter case, the position can be assigned to a specific line (first/middle/last/chosen). Moreover, equation numbers can be turned on and off by commands, and they can be triggered by setting a label.
- The vertical spacing above and below single- and multi-line equations of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and `amsmath` can be somewhat variable, hard to control and even resistive in certain situations. The package implements clearer structures controlling the vertical spacing, including proper dependency on the text line above and ways to adjust the spacing.

- The framework introduces a scheme which semi-automatically inserts punctuation, e.g. ‘.’ or ‘;’, at the end of the following (or every) equation environment. Punctuation can also be inserted at every alignment column or equation line including the possibility to prepend a certain spacing.
- Next to `\[...]` as an alias for the single-line `equation` environment, the package uses `\langle... \rangle` as an alias multi-line equations.
- The horizontal alignment and indentation of equation lines can be adjusted via a scheme or on a line-by-line basis.
- The alignment marker can be placed before or after the equation signs while maintaining proper spacing to symbols before and after it. This simplifies the construction of continuing equations in an aligned context.
- Equation lines are subject to shrinking of white space if the available space does not suffice (analogously to single-line equations).
- Most settings can be controlled via optional arguments and modifiers to the equation environment or via global settings. This includes switching between different types of equation environments, enabling or disabling numbering, adjusting vertical spacing, etc. This feature simplifies the adjustment and fine-tuning of equations towards the intended layout.
- Last but not least, the underlying `amsmath` code, originating from the `TEX` era and early `LATEX` years, has been redesigned with emphasis on clarity, readability, adjustability and maintainability (but at the cost of moderately higher resource consumption and moderately lower efficiency). Nevertheless, it remains `LATEX 2ε` code.

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

2 Usage

Notice regarding package version v0.5: 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

```
\usepackage{eqnlines}
```

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

2.1 Equations Environment

`equations` (*env.*) The package supplies a main maths environment called `equations` which accepts a comma-separated list of optional parameters ‘`[opts]`’:

```
\begin{equations}[opts]_
...
\end{equations}
```

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

We note that the `equations` environment should be started with a white space character ‘`_`’ which provides a clear separation from optional arguments ‘`[opts]`’ and/or modifiers which must immediately follow the environment declaration `\begin{equations}`.

`single` (*key*) The environment has three principal modes of operation which 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, eq, 1</code>	<code>gather, ga, ln, ~</code>	<code>align, al, col, @</code>
symbolic	<code>\[...\]</code>	<code>\<~...\></code>	<code>\<...\></code>
<code>amsmath</code> env.	<code>equation</code>	<code>gather, multiline</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).

`\[...\]` The package offers several alternative names for the same mode as well as a symbolic short `\<...\>` form extending the \LaTeX display equation form `\[...\]` to multi-line equations.

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

`sqropt` (*key*) optional argument `lines` selecting the lines mode. Both short forms can be customised by

`angopt` (*key*) setting default arguments via the global options `sqropt={opts}` and `angopt={opts}`. Both default arguments are preset to `nonumber` which disables equation numbering, see section 2.2.

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

`multiline` (*env.*) `gather`, `align` and `flalign` including their starred variants but neither the `alignat` alter-

`gather` (*env.*) native forms nor `split`. It is possible to define further equation environments *env* with a

`align` (*env.*) predefined set of options *opts* using:

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

2.2 Numbering

`numberline` (*key*) The package extends the established interface of L^AT_EX and the `amsmath` package for labelling equations with numbers or with manually assigned tags. For multi-line equations, there are two distinct modes of operations: individual labelling of the equation lines or one overall number/tag for the whole block of equations. The modes are selected by an optional argument `numberline=mode` as follows:

name	alt.	description	preset
<code>none</code>	<code>n</code>		preset off
<code>all</code>	<code>a</code>	individual lines	preset on
<code>sub</code>	<code>s</code>		subequations (a, b, c, ...)
<code>first</code>	<code>f</code>		first line
<code>last</code>	<code>l</code>		last line
<code>middle</code>	<code>m</code>	single number	middle line
<code>out</code>	<code>o</code>		last/first line for right/left tags
<code>in</code>	<code>i</code>		first/last line for right/left tags
<code>here</code>	<code>h</code>		line indicated by <code>\numberhere</code>

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

`\nonumber` and `\donumber`

* (*key*) Alternatively, the numbering can be disabled or enabled for the block using modifiers (which ! (*key*) must be placed *before* further optional arguments):

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

This allows to define a default behaviour and specify exceptions where they may occur. The star modifier following directly the environment declaration replaces the starred form of environments (`equation*`, etc.).

`\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, e.g. if an equation is broken into several lines and the last one should receive the number tag.

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

`\label{label}` and `\tag[*]{tag}`

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 for the present line or block (`\donumber`). By default, numbers are triggered for `\tag`, but not for `\label` reflecting the behaviour set forth by `amsmath`. By enabling triggering for `\label`, numbers will be produced only if they have a chance of being referenced.

`label` (*key*) The equations environment provides an alternative means to specify labels and tags within
`tag` (*key*) the optional arguments [*opts*] or via the modifier `@{label}` (which may *follow* further op-
`@` (*key*) tional arguments):

`label={label}`, `tag[*]={tag}`, `\[0{label}...\]`

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.

`\raisetag`

`taglayout (key)` The typesetting of equation numbers and tags passes through two macros, one which defines the layout and another one which adds a decoration (brackets). These two methods can be adjusted via the options:

`taglayout[*]={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 brackets for the tag presentation.

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

2.3 Horizontal Adjustment

`margin (key)` The horizontal alignment of columns is fixed for aligned multi-line equations: Each pair of subsequent columns forms a unit which is aligned at the intermediate alignment marker ‘&’. These columns are distributed evenly over the available horizontal space. Here, the outer space left and right of the set of columns can be treated on equal footing to the space between the columns (option `margins=on`), but it can be eliminated so that the outer columns are pushed right to the margin (option `margins=off`). In addition, a minimum and maximum width can be specified for the column separation (`mincolsep=dimen` and `maxcolsep=dimen`). By default, no maximum column separation is set (`maxcolsep*`), and all horizontal space is used, otherwise a value of `2em` ($\equiv\backslashqqquad$) is suggested with the minimum separation set to `1em` ($\equiv\backslashquad$) by default.

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

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

Note that the `steps` shape comes to use in the `amsmath` environment `multline`. The alignment preset can be adjusted for individual lines by the macros:

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

In contradistinction to `amsmath`, these macros do not require to specify the cell contents as their argument (but there is no harm in doing so). The macro `\shoveleft` also accepts the modifiers ‘*’ or ‘!’ for indentation or hanging indentation by the standard indentation amount (`indent=2em`) or an optional argument [`indent`] specifying a variable amount of indentation.

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

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

`left` (*key*) In central alignment mode, there is the option of imposing a tag margin `tagmargin` = *dimen* which allocates some space to the tag such that equation content is centred in the remaining horizontal space.

`leftmargin` (*key*) In left alignment mode, all equations are left aligned to a left margin (`leftmargin` initialised

`minleftmargin` (*key*) to the first level of enumerations and itemisations). Depending on the situation, the left

`maxleftmargin` (*key*) margin may be reduced or extended to `minleftmargin` or `maxleftmargin`, respectively.

`marginbadness` (*key*) Finally, we note that within single and stacked equations, very long equations that do not

`maxbadness` (*key*) fit the available horizontal space are subject to shrinking attempts. In other words, \TeX will attempt to shrink the glue contained in the equation line to make it fit. This shrinking can be controlled by the two parameters `marginbadness` and `maxbadness` accepting integer values. The former is used for trying to shrink onto certain horizontal margins which are otherwise reserved for tags; the latter is used for using the maximal horizontal space which also raises or lowers the equation tag if needed. Small values prevent shrinking and higher values allow for more compression.

2.4 Punctuation

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

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

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

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

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

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

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

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 \TeX '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.

`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` determines the math class just after the alignment if the cell before alignment is empty.

For example, the following two expressions produce identical output:

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

2.6 Vertical Spacing

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

The spacing of equations to the surrounding text is a combination of several aspects:

First, \TeX inserts some interline spacing according to its rules. The amount depends on the depth/height of the surrounding text and the height/depth of the math content. The former typically takes rather uniform values, whereas the latter can range wildly with the context (plain equations vs. fractions and matrices). As equations are normally surrounded by a relatively large amount of glue, it makes sense to reduce the dependency on the height/depth of math content. Therefore, the package makes equation environments appear to the surrounding text as a line with a fixed height and depth, and thus interline glue merely fills

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

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. 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:

`shortmode` (*key*)

name	reduced glue
<code>off</code>	disabled
<code>above</code>	above short equations
<code>belowone</code>	also below short single-line equations
<code>belowall</code>	also below all short single-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`.

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

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

`par` (*key*) The key `par` controls to whether the equations environment end in horizontal mode (value `cont`) or in vertical mode (value `par`, default) with a dedicated amount of glue `belowparskip`. An environment can also end in vertical mode without interline skip (value `top`) and with 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
`...space` (*key*) bodies where it adds vertical space below the next equation line or below the equation environment. Additional glue can be added above or below equation environments by means of the options `abovespace` and `belowspace`.

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

<code>...posskip</code>	both	description
<code>...long...</code>	<code>skip</code>	standard amount of glue
<code>...short...</code>	–	reduced glue for short equations
<code>...cont...</code>	–	glue when issued from an empty <code>\noindent</code> paragraph
<code>...par...</code>	–	glue when starting a paragraph (in vertical mode)
<code>...top...</code>	–	glue when issued at the top of vertical list
<code>...med...</code>	<code>medskip</code>	medium amount of glue
<code>...tag...</code>	<code>tagskip</code>	glue for outer raised/lowered tags
<code>...medtag...</code>	<code>medtagskip</code>	glue for outer raised/lowered tags with med
<code>...partag...</code>	–	glue for outer raised/lowered tags with par

`...mode` (*key*) The situations `cont`, `par` and `top` use the respective glue `above...skip` above the equations. By default they use `belowlongskip` below the equations, but this setting may be adjusted by the global option `posmode` with the values:

name	reduced glue
<code>long</code>	regular amount below equations
<code>short</code>	short amount below equations
<code>cont</code>	<code>cont</code> amount below equations
<code>par</code>	end the paragraph (only <code>parmode</code> and <code>topmode</code>)
<code>top</code>	end the paragraph without interline skip (<code>topmode</code> only)

`spread` (*key*) Likewise, the spacing between the lines of a multi-line equation environment can be adjusted via `spread={dimen}` which defaults to `\jot=3pt`. In addition, all equation lines and tags

`strut` (*key*) are supplied with struts to ensure a minimum height and depth. The latter behaviour is controlled by the boolean switches `strut` and `strutttag`.

`displaybreak` (*key*) Finally, the breaking of multi-line equations across pages can be controlled as follows: The `\displaybreak` setting `allowdisplaybreaks` taking values 0 (never) through 4 (permissive) controls the per-

`owdisplaybreaks` (*key*) mittivity of page breaks within multi-line equations. The optional argument `displaybreak` taking values 0 (do not) through 4 (enforce) suggests a break just *above* the equation environment. The command `\displatybreak` with values 0 through 4 suggests a break below the current line or below the equation environment.

2.7 Further Environments

The package supplies some additional environments:

`equationsbox` (*env.*) The package provides a boxed equation environment `equationsbox` which can be used within arbitrary math content. It works analogously to `equations` including optional arguments

`margin` (*key*) and modifiers, but it offers a reduced range of functionality such as (evidently) no numbering.

`marginleft` (*key*) Additional arguments are given by `margin`, `marginleft`, `marginright` which specify additional margin space around the equations box.

`marginright` (*key*)

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

```
subeqtemplate={\theparentequation\alph{equation}}
```

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

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 to allow special commands (e.g. <code>\verb</code>)
<code>linesfallback</code>	single column in align mode reverts to lines mode
<code>ampproof</code>	equip optional argument parsing with protection for <code>&</code>
<code>crerror</code>	invoke an error when ‘ <code>\</code> ’ is used in a single equation

2.9 Feature Selection and Package Options

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

option	type	description
<code>equation</code>	bool	provide/overwrite <code>equation</code> and <code>\[...\]</code>
<code>amsmath</code>	bool	provide/overwrite <code>amsmath</code> environments and macros
<code>ang</code>	bool	provide <code>\<...\></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:

$$\backslash eqnlinesprovide\{features\}$$

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>
<i>cmd</i>	provide/overwrite macro <code>\cmd</code> : <code>eqref</code> , <code>notag</code> , <code>thetag</code> , <code>allowdisplaybreaks</code> , <code>numberwithin</code>
<code>tagform</code>	provide/overwrite macro <code>\tagform@</code>
<code>maketag</code>	provide/overwrite macro <code>\maketag@@@</code>

3 Information

3.1 Copyright

Copyright © 2024–2025 Niklas Beisert

Based on the latex package `amsmath`: Copyright © 1995, 2000, 2013 American Mathematical Society; 2016–2024 LaTeX Project and American Mathematical Society.

This work may be distributed and/or modified under the conditions of the L^AT_EX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in <https://www.latex-project.org/lppl.txt> and version 1.3c or later is part of all distributions of L^AT_EX version 2008 or later.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is Niklas Beisert.

This work consists of the files `README.txt`, `eqnlines.ins` and `eqnlines.dtx` as well as the derived files `eqnlines.sty` and `eqnlines.pdf`.

3.2 Credits

This package is based on the L^AT_EX package `amsmath` (initially named `amstex`) which in turn is based on the T_EX macro system `amstex` written by Michael Spivak. The initial work of porting `amstex` to L^AT_EX was done in 1988–1989 by Frank Mittelbach and Rainer Schöpf. In

1994 David M. Jones added the support for flush-left layout and did extensive improvements to the align family of environments and to the equation number handling in general. Michael Downes at the AMS served as coordinator for the efforts of Mittelbach, Schöpf, and Jones, and has contributed various bug fixes and additional refinements over time. Since 2016, the package has been maintained by the LaTeX Project with contributions by the above and David Carlisle.

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

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

3.3 Files and Installation

The package consists of the files:

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

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

- Run `(pdf)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` will rename the math environments of `amsmath` with a prefix `ams...` and overwrite them with its own implementations. Alternatively, `eqnlines` may assign individual names to the maths environments and preserve the ones of `amsmath`. The other features provided by `amsmath` can be used.
- The package `mathtools` is a popular extension of the `amsmath` package. This package incorporates some of the features and improvements provided by the `mathtools` package. Compatibility with the `mathtools` package has been tested with v1.31 (2024/10/04), and it is maintained whether `eqnlines` is loaded before or after `mathtools`. Some features like adding a box and emphasising equations via `empheq` does not (yet) work.
- This package cooperates with the package `hyperref` to create anchors and references within the electronic document. Compatibility with the `hyperref` package has been tested with v7.011 (2024/11/05).
- This package supports the display of labels and references through the package `showkeys`. Compatibility with the `showkeys` package has been tested with v3.21 (2024/05/23).

3.5 Feature Suggestions

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

- a proper manual
- complete code documentation
- sample document

3.6 Revision History

v0.6: 2025/03/11

- preliminary pdf tagging support (<https://latex3.github.io/tagging-project/>)
- classic L^AT_EX/`amsmath` vs. `eqnlines` presets
- changed vertical spacing schemes and added further options
- supplied dimensions processed by `\glueexpr`
- more independent of `amsmath` structures
- internal reorganisations

v0.5: 2025/02/25

- preview version published on CTAN

A Implementation

The appendix documents the various components of the present package.

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

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

B General Support

In the following we describe general purpose supporting routines.

B.1 Development Messages

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

```

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

B.2 Look-Ahead in Alignment

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

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

```

8 \let\eql@ifnextchar@loose\kernel@ifnextchar
9 \long\def\eql@ifnextchar@tight#1#2#3{%
10 \let\reserved@d= #1%
11 \def\reserved@a{#2}%
12 \def\reserved@b{#3}%
13 \futurelet\@let@token\eql@ifnch@tight
14 }
15 \def\eql@ifnch@tight{%
16 \ifx\@let@token\reserved@d
17 \let\reserved@b\reserved@a
18 \fi
19 \reserved@b
20 }
```

TODO: describe

```
21 \begingroup
22 \makeatother
23 \let\tmp=%
24 \makeatletter
25 \global\let\eql@atxii\tmp
26 \endgroup
```

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

```
27 \long\def\eql@ifnextgobble@loose#1#2{\eql@ifnextchar@loose#1{\@firstoftwo{#2}}}
28 \long\def\eql@ifnextgobble@tight#1#2{\eql@ifnextchar@tight#1{\@firstoftwo{#2}}}
29 \long\def\eql@ifstar@loose#1{\eql@ifnextchar@loose*{\@firstoftwo{#1}}}
30 \long\def\eql@ifstar@tight#1{\eql@ifnextchar@tight*{\@firstoftwo{#1}}}
31 \long\def\eql@ifat@loose#1#2{\eql@ifnextgobble@loose{#1}{%
32 \eql@ifnextgobble@loose\eql@atxii{#1}{#2}}}
33 \long\def\eql@ifat@tight#1#2{\eql@ifnextgobble@tight{#1}{%
34 \eql@ifnextgobble@tight\eql@atxii{#1}{#2}}}
35 \long\def\eql@testopt@loose#1#2{\eql@ifnextchar@loose[#{1}]{#1[#{2}]}%]
36 \long\def\eql@testopt@tight#1#2{\eql@ifnextchar@tight[#{1}]{#1[#{2}]}%]
37 \long\def\eql@teststaropt@loose#1#2#3{%
38 \eql@ifstar@loose{\eql@testopt@loose{#1}{#3}}{\eql@testopt@loose{#2}{#3}}}
39 \long\def\eql@teststaropt@tight#1#2#3{%
40 \eql@ifstar@tight{\eql@testopt@tight{#1}{#3}}{\eql@testopt@tight{#2}{#3}}}
```

`\eql@spbgroup` The second challenge is addressed by enclosing the look-ahead in spurious groups¹ which protect against triggering ‘&’. The macros `\eql@spbgroup` and `\eql@spegroup` open and close a spurious group. For some reason, the look-ahead mechanism requires further protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end. These adjustments are included in the macros `\eql@srbggroup` and `\ers@spegroup`:

```
41 \def\eql@spbgroup{\iffalse{\fi\ifnum0='}\fi}
42 \def\eql@spegroup{\ifnum0='{ \fi\iffalse}\fi}
43 \def\eql@srbggroup{\relax\iffalse{\fi\ifnum0='}\fi}
44 \def\eql@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}
```

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

```
45 \long\def\eql@ampprotect#1#2{\eql@srbggroup#1{\eql@sregroup#2}}
46 \long\def\eql@ampprotecttwo#1#2#3{%
47 \eql@srbggroup#1{\eql@sregroup#2}{\eql@sregroup#3}}
```

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

```
48 \def\eql@ifnextchar@loose@ampsafe#1{%
49 \eql@ampprotecttwo\eql@ifnextchar@loose#1}}
50 \def\eql@ifnextchar@tight@ampsafe#1{%
51 \eql@ampprotecttwo\eql@ifnextchar@tight#1}}
```

¹See <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>.

```

52 \def\eq@ifstar@loose@ampsafe{\eq@ampprotecttwo\eq@ifstar@loose}
53 \def\eq@ifstar@tight@ampsafe{\eq@ampprotecttwo\eq@ifstar@tight}
54 \def\eq@testopt@loose@ampsafe{\eq@ampprotect\eq@testopt@loose}
55 \def\eq@testopt@tight@ampsafe{\eq@ampprotect\eq@testopt@tight}
56 \def\eq@teststaropt@loose@ampsafe{\eq@ampprotecttwo\eq@teststaropt@loose}
57 \long\def\eq@teststaropt@tight@ampsafe{%
58   \eq@ampprotecttwo\eq@teststaropt@tight}

```

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

```

59 \let\eq@ifnextchar@org\@ifnextchar
60 \let\eq@ifstar@org\@ifstar
61 \let\eq@testopt@org\@testopt
62 \def\eq@amprevert{%
63   \let\@ifnextchar\eq@ifnextchar@org
64   \let\@testopt\eq@testopt@org
65   \let\@ifstar\eq@ifstar@org
66 }
67 \def\eq@ampproof{%
68   \let\@ifnextchar\eq@ifnextchar@loose@ampsafe
69   \let\@testopt\eq@testopt@loose@ampsafe
70   \let\@ifstar\eq@ifstar@loose@ampsafe
71 }

```

B.3 Error Messages

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

```

\eq@warning
72 \def\eq@error#1{\PackageError{eqnlines}{#1}-{}}
73 \def\eq@warning{\PackageWarning{eqnlines}}

```

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

```

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

```

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

```

\warn@label@multiple
\eq@warn@tag@unused
\warn@tag@multiple
76 \def\eq@warn@label@unused{\eq@warning{Unused equation \string\label:
77   label '\eq@nextlabel' will be lost}}
78 \def\eq@warn@label@multiple#1{\eq@warning{Multiple equation \string\label's:
79   previous label '#1' will be lost}}
80 \def\eq@warn@tag@unused{\eq@warning{Unused equation \string>tag:
81   tag declaration will be lost}}
82 \def\eq@warn@tag@multiple{\eq@warning{Multiple equation \string>tag's:
83   previous tag declaration will be lost}}

```

B.4 amsmath Integration

`\eq@amsmath@after` We need to overwrite certain macros from `amsmath`. The method `\eq@amsmath@after` executes argument #1 after loading `amsmath` is loaded. It also runs the code if `amsmath` has already been loaded. Furthermore, loading `amsmath` requires certain macros to be undefined. To this end `\eq@amsmath@before` will execute argument #1 before any future

loading of `amsmath`. `\eql@amsmath@undefine` undefines a macro in this way and `\eql@amsmath@let` overwrites a macro of `\ctanpkg{amsmath}`:

```
84 \def\eql@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
85 \def\eql@amsmath@before#1{%
86   \ifpackageloaded{amsmath}{}\AddToHook{package/amsmath/before}{#1}}
87 \def\eql@amsmath@undefine#1{\eql@amsmath@before{\let#1\undefined}}
88 \def\eql@amsmath@let#1#2{\eql@amsmath@undefine#1\let#1#2}
```

C Parameters and Registers

In the following, we collect parameter and register definitions.

C.1 Supporting Definitions

`\eql@false` (*bool*) Rather than the standard \LaTeX 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 - \TeX conditional `\ifdefined\xxx`. In order to make the scheme more tangible, we define the two expected values for boolean variables:

```
89 \let\eql@false\undefined
90 \let\eql@true\empty
```

`\eql@regunset@` (*skip*) Some dimension registers need to be initialised to values which may depend on fonts and styles, e.g. `1em`. As `\@mathmargin` of `amsmath`, we use the special value `-1sp` to indicate that this register should be initialised at the end of the preamble, and we define a skip register to hold this value. The macro `\eql@regsetdelayed` performs the delayed initialisation:

```
91 \newskip\eql@regunset@
92 \eql@regunset@-1sp\relax
93 \def\eql@regsetdelayed#1#2{%
94   #1\eql@regunset@\relax
95   \AtBeginDocument{%
96     \ifdim#1=\eql@regunset@
97       #1\glueexpr#2\relax
98     \fi
99   }%
100 }
```

C.2 Parameters

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:

```
101 \let\eql@tagsleft\eql@false
```

`\eql@flushleft` (*bool*) The boolean parameter `\eql@flushleft` specifies whether the layout is flush left or centered:

```
102 \let\eql@flushleft\eql@false
```

`\flushleftmargin@` (*dimen*) The default width of the left margin in flush-left mode is specified by `\flushleftmargin@`. It may be pushed down to `\flushleftmarginmin@` and up to `\flushleftmarginmax@`:

```
103 \newdimen\eq\flushleftmargin@
104 \newdimen\eq\flushleftmarginmin@
105 \newdimen\eq\flushleftmarginmax@
106 \eq\regsetdelayed\eq\flushleftmargin@\leftmargini
107 \eq\flushleftmarginmin@\z@
108 \eq\flushleftmarginmax@.5\maxdimen
```

`\tagmargin@` (*dimen*) The intended margin width for tags in centered layout is specified by `\tagmargin@`:

```
109 \newdimen\eq\tagmargin@
110 \eq\tagmargin@\z@
```

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

```
111 \newdimen\eq\indent@
112 \def\eq\indent@val{2em}
```

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

```
113 \newdimen\eq\paddingleft@
114 \newdimen\eq\paddingright@
115 \def\eq\paddingleft@val{0pt}
116 \def\eq\paddingright@val{0pt}
```

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

```
117 \let\eq\paddingmax\eq>false
```

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

```
118 \def\eq\box@marginleft{\z@skip}
119 \def\eq\box@marginright{\z@skip}
```

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

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

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

```
121 \def\eq\spread{\jot}
122 \newdimen\eq\spread@amount@
```

`\tagfuzz@` (*dimen*) The value `\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.

```
123 \newdimen\eq\tagfuzz@
124 \eq\tagfuzz@16sp\relax
```

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

```
125 \def\eql@display@height{\ht\eql@strutbox@}
126 \def\eql@display@depth{\dp\eql@strutbox@}
```

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

```
127 \newcount\eql@skip@mode@short@
128 \eql@skip@mode@short@\tw@

129 \newcount\eql@skip@mode@cont@
130 \eql@skip@mode@cont@\z@

131 \newcount\eql@skip@mode@par@
132 \eql@skip@mode@par@\z@

133 \newcount\eql@skip@mode@top@
134 \eql@skip@mode@top@\z@

135 \newcount\eql@skip@mode@leave@
136 \let\eql@skip@force@leave@\undefined
```

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

```
\eql@skip@force@below
\mode@above@ (counter)
\mode@below@ (counter)
137 \newcount\eql@skip@mode@above@
138 \newcount\eql@skip@mode@below@
139 \let\eql@skip@force@above@\undefined
140 \let\eql@skip@force@below@\undefined
141 \let\eql@skip@custom@above@\undefined
142 \let\eql@skip@custom@below@\undefined
```

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

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

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

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

```
143 \def\eql@skip@long@above{\abovedisplayskip}
144 \def\eql@skip@long@below{\belowdisplayskip}
145 \def\eql@skip@short@above{\abovedisplayshortskip}
146 \def\eql@skip@short@below{\belowdisplayshortskip}
147 \def\eql@skip@cont@above{\eql@skip@short@above}
148 \def\eql@skip@cont@below{\eql@skip@short@below}
149 % \TODO: parabove plus parskip?
150 \def\eql@skip@par@above{\eql@skip@long@above}
151 \def\eql@skip@par@below{\eql@skip@long@below}
152 \def\eql@skip@top@above{\eql@skip@long@above}
```

```

153 \def\eq@skip@top@below{\eq@skip@long@below}
154 \def\eq@skip@med@above{\abovedisplayskip/2}
155 \def\eq@skip@med@below{\belowdisplayskip/2}
156 \def\eq@skip@tag@above{\z@skip}
157 \def\eq@skip@tag@below{\z@skip}
158 \def\eq@skip@partag@above{\z@skip}
159 \def\eq@skip@partag@below{\z@skip}
160 \def\eq@skip@medtag@above{\z@skip}
161 \def\eq@skip@medtag@below{\z@skip}

```

`\eq@colsepmin@` (*dimen*) The minimum and maximum intercolumn separation is specified by `\eq@colsepmin@` and `\eq@colsepmax@`. These dimension registers are set to the macros `\eq@colsepmin@val` and `\eq@colsepmax@val`, respectively, when entering the equation environments to allow font-dependent values including calc evaluations:

```

162 \newdimen\eq@colsepmin@
163 \newdimen\eq@colsepmax@
164 \def\eq@colsepmin@val{1em}
165 \def\eq@colsepmax@val{.5\maxdimen}

```

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

```

166 \newdimen\eq@tagwidthmin@
167 \eq@tagwidthmin@\z@

```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. TeX's built-in value is half a quad in font number 2 (`.5\fontdimen6\textfont\tw@`). As the tag is processed in text mode, we use `0.5em` instead. **TODO:** may set within environment

```

168 \newdimen\eq@tagsepmin@
169 \eq@regsetdelayed\eq@tagsepmin@{.5em}

```

C.3 Registers

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

```

170 \newcount\eq@row@
171 \newcount\eq@totalrows@

```

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

```

172 \newdimen\eq@colsep@

```

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

```

173 \newdimen\eq@display@firstavail@
174 \def\eq@display@firstavail@set#1{%
175   \ifnum\eq@row@=\@ne
176     \global\eq@display@firstavail@#1%
177   \fi
178 }

```

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

```
179 \newcount\eq@raisetag@firstlast@
```

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

`\eq@tagbox@` (*box*) tag for the present line. The corresponding dimensions `\eq@fieldwidth@` and

`\eq@fieldwidth@` (*dimen*) `\eq@tagwidth@` hold their widths:

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

```
180 \newbox\eq@fieldbox@
181 \newbox\eq@tagbox@
182 \newdimen\eq@fieldwidth@
183 \newdimen\eq@tagwidth@
```

`\eq@line@height@` (*dimen*) The dimension registers `\eq@line@height@` and `\eq@line@depth@` keep track of the

`\eq@line@depth@` (*dimen*) height and depth of the present line in an alignment:

```
184 \newdimen\eq@line@height@
185 \newdimen\eq@line@depth@
```

`\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:

```
186 % \newif\ifmeasuring@
187 \ifdefined\measuring@true\else
188 \expandafter\newif\csname ifmeasuring@\endcsname
189 \fi
190 \AddToHook{package/hyperref/after}{
191 \ifdefined\Hy@ifnotmeasuring
192 \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
193 \fi
194 }
```

`\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:

```
195 % \newif\if@display
196 \ifdefined\@displaytrue\else
197 \expandafter\newif\csname if@display\endcsname
198 \everydisplay\expandafter{\the\everydisplay\@displaytrue}
199 \fi
```

C.4 Hooks

`\eq@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

```
200 \let\eq@hook@blockbefore\@empty
201 \let\eq@hook@blockafter\@empty
202 \let\eq@hook@blockin\@empty
```

```

203 \let\eqL@hook@blockout\@empty
204 \let\eqL@hook@linein\@empty
205 \let\eqL@hook@lineout\@empty
206 \let\eqL@hook@colin\@empty
207 \let\eqL@hook@colout\@empty
208 \let\eqL@hook@eqin\@empty
209 \let\eqL@hook@eqout\@empty
210 \let\eqL@hook@innerleft\@empty
211 \let\eqL@hook@innerright\@empty
212 \let\eqL@hook@innerlead\@empty

```

D Tagging Support

```

213 \let\eqL@tagging@on\eqL@false
214 \IfFormatAtLeastTF{2025-06-01}{%
215   \csname tag_if_active:T\endcsname{\let\eqL@tagging@on\eqL@true}}{}
216 \ifdefined\eqL@tagging@on
217   \def\eqL@tagging@mathsave{%
218     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
219   \def\eqL@tagging@mathaddlast{%
220     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
221   \def\eqL@tagging@tagbegin{%
222     \UseTaggingSocket{math/display/tag/begin}}
223   \def\eqL@tagging@tagend{%
224     \UseTaggingSocket{math/display/tag/end}}
225   \def\eqL@tagging@tagsave{%
226     \UseTaggingSocket{math/luamml/mtable/tag/save}}
227   \def\eqL@tagging@tagaddbox{%
228     \setbox\z@\copy\eqL@tagbox%
229     \UseTaggingSocket{math/luamml/mtable/tag/set}}
230   \def\eqL@tagging@tablesaverinner{%
231     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
232   \def\eqL@tagging@tableaddinner{%
233     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
234   \def\eqL@tagging@tablesavelines{%
235     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
236   \def\eqL@tagging@tablesavealign{%
237     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
238   \def\eqL@tagging@alignleft{%
239     \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
240   \def\eqL@tagging@aligncenter{%
241     \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
242   \def\eqL@tagging@alignright{%
243     \UseTaggingSocket{math/luamml/mtable/aligncol}{right}}
244   \let\eqL@single@doscan\eqL@true
245   \let\eqL@scan@body\eqL@scan@body@rescan
246   \ExplSyntaxOn
247   \def\eqL@tagging@start{%
248     \bool_if:NF\l__math_collected_bool{%
249       \toks@\expandafter{\eqL@tagging@opt}%
250       \edef\eqL@tmp{\@currenvr}{[\the\toks@] \the\eqL@scan@reg}}}%
251     \expandafter\__math_process:nn\eqL@tmp
252     \@kernel@math@registered@begin
253     \bool_set_true:N\l__math_collected_bool
254   }%
255 }
256 \def\eqL@tagging@end{}

```

```

257 \ExplSyntaxOff
258 \else
259 \def\eql@tagging@mathsave{}
260 \def\eql@tagging@mathaddlast{}
261 \def\eql@tagging@tagbegin{}
262 \def\eql@tagging@tagend{}
263 \def\eql@tagging@tagsave{}
264 \def\eql@tagging@tagaddbox{}
265 \def\eql@tagging@tablesaveinner{}
266 \def\eql@tagging@tableaddinner{}
267 \def\eql@tagging@tablesavealign{}
268 \def\eql@tagging@tablesavealign{}
269 \def\eql@tagging@alignleft{}
270 \def\eql@tagging@aligncenter{}
271 \def\eql@tagging@alignright{}
272 \def\eql@tagging@start{}
273 \def\eql@tagging@end{}
274 \fi

```

E 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 `\relax` indicates that the punctuation should be handed down to the next lower level:

```

\eql@punct@line \relax
\eql@punct@block
275 \let\eql@punct@col\empty
276 \let\eql@punct@line\relax
277 \let\eql@punct@block\relax

```

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

```

278 \let\eql@punct@sep\relax

```

`\eqnpunctcol` Set the punctuation for columns, lines and blocks. Note that the macro `\eqnpunct` sets the punctuation for the next equation block (only). Starred versions clear the punctuation for the respectively levels:

```

\eqnpunct
279 \def\eqnpunctcol{\eql@ifstar@tight\eql@punct@col@setrelax\eql@punct@col@set}
280 \def\eql@punct@col@set#1{\def\eql@punct@col{#1}\ignorespaces}
281 \def\eql@punct@col@setrelax{\let\eql@punct@col\empty\ignorespaces}
282 \def\eqnpunctline{\eql@ifstar@tight\eql@punct@line@setrelax\eql@punct@line@set}
283 \def\eql@punct@line@set#1{\def\eql@punct@line{#1}\ignorespaces}
284 \def\eql@punct@line@setrelax{\let\eql@punct@line\relax\ignorespaces}
285 \def\eqnpunctmain{\eql@ifstar@tight\eql@punct@main@setrelax\eql@punct@main@set}
286 \def\eql@punct@main@set#1{\eqnlineset{punct=#1}\ignorespaces}
287 \def\eql@punct@main@setrelax{\eqnlineset{punct*}\ignorespaces}
288 \def\eqnpunct{\eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
289 \def\eql@punct@next@set#1{\eqnaddopt{punct=#1}\ignorespaces}
290 \def\eql@punct@next@setrelax{\eqnaddopt{punct*}\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:

```

291 \def\eql@punct@apply@col{%

```

```

292 \ifx\eqlopunct@col\@empty\else
293   \eqlopunct@sep
294   \eqlopunct@col
295   \let\eqlopunct@col\@empty
296 \fi
297 }

```

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

`\eqlopunct@apply@line`

```

298 \def\eqlopunct@apply@line{%
299   \ifx\eqlopunct@line\relax
300 % \TODO hand down immediately?
301   \else
302     \ifx\eqlopunct@line\@empty\else
303       \eqlopunct@sep
304       \eqlopunct@line
305     \fi
306     \let\eqlopunct@line\relax
307     \let\eqlopunct@col\@empty
308   \fi
309 }

```

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

```

310 \def\eqlopunct@apply@block{%
311   \ifx\eqlopunct@block\relax
312 % \TODO hand down immediately?
313   \else
314     \ifx\eqlopunct@block\@empty\else
315       \eqlopunct@sep
316       \eqlopunct@block
317     \fi
318     \let\eqlopunct@block\relax
319     \let\eqlopunct@line\relax
320     \let\eqlopunct@col\@empty
321   \fi
322 }

```

F Math Classes at Alignment

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

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

```

323 \def\eqlopclass@innerright@sel@{%
324   \ifdim\eqlopfieldwidth@=\z@
325     \eqlopclass@innerlead
326   \else
327     \eqlopclass@innerright
328   \fi
329 }

```

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

```

\eq@class@innerlead@set
330 \def\eql@class@innerleft@set#1{%
331   \def\eql@class@innerleft{#1}%
332 }
333 \def\eql@class@innerright@set#1{%
334   \def\eql@class@innerright{#1}%
335   \let\eql@class@innerright@sel\eql@class@innerright
336 }
337 \def\eql@class@innerlead@set#1{%
338   \def\eql@class@innerlead{#1}%
339   \let\eql@class@innerright@sel\eql@class@innerright@sel@
340 }

```

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

```

341 \def\eql@class@ampeq{%
342   \eql@class@innerleft@set{}%
343   \eql@class@innerright@set{{}}%
344 }
345 \def\eql@class@eqamp{%
346   \eql@class@innerleft@set{\mathrel{}}%
347   \eql@class@innerright@set{\mathrel{}}%
348   \eql@class@innerlead@set{{}}%
349 }
350 \eql@class@ampeq

```

G Equations Box Environment

TODO: describe

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

`\eql@box@cr`

```

351 \protected\def\eql@box@cr{%
352   \eql@ampprotect\eql@testopt@tight\eql@box@cr@z@
353 }
354 \def\eql@box@cr#[#1]{%
355   \eql@punct@apply@line
356   \eql@hook@lineout
357   \eql@box@lastfield
358   \cr
359   \noalign{%
360     \vskip\glueexpr#1\relax
361   }%
362 }

363 \let\eql@box@box\vcenter

364 \def\eql@box@lastfield@odd{%
365   &\omit
366   \kern-\wd\eql@fieldbox@
367   \box\eql@fieldbox@
368   \hfil
369   &\omit\kern-\eql@colsep@

```

```

370 }%
371 \def\eql@box@lastfield@even{&\omit\kern-\eql@colsep@}
372 \def\eql@box@lastfield@lines{&\omit\kern-2\eql@colsep@}

373 \def\eql@box@open@align{%
374 % \TODO templates
375 \let\eql@box@lastfield@empty
376 \eql@halign@init{%
377 (dev)\eql@dev{starting new line}%
378 }%
379 \tabskip\z@skip
380 \halign\bgroup
381 &%
382 \let\eql@box@lastfield\eql@box@lastfield@odd
383 \global\setbox\eql@fieldbox@\hbox{%
384 \eql@strut@field
385 \@lign
386 $\m@th\displaystyle
387 \eql@hook@colin
388 ##%
389 \eql@class@innerleft
390 \eql@hook@innerleft
391 \eql@tagging@mathsave
392 $%
393 \eql@tagging@mathaddlast
394 }%
395 \hfil
396 \kern\wd\eql@fieldbox@
397 \tabskip\z@skip
398 &%
399 \eql@fieldwidth@\wd\eql@fieldbox@
400 \kern-\eql@fieldwidth@
401 \box\eql@fieldbox@
402 \let\eql@box@lastfield\eql@box@lastfield@even
403 \llap{\unhbox\eql@fieldbox@}%
404 \hbox{%
405 \eql@strut@field
406 \@lign
407 $\m@th\displaystyle
408 \eql@hook@innerright
409 \eql@class@innerright@sel
410 ##%
411 \eql@punct@apply@col
412 \eql@hook@colout
413 \eql@tagging@mathsave
414 $%
415 \eql@tagging@mathaddlast
416 }%
417 \hfil
418 \tabskip\eql@colsep@\relax
419 \crrc
420 \noalign{%
421 \eql@hook@blockbefore
422 }%
423 \eql@hook@blockin
424 }

425 \def\eql@box@open@lines{%
426 % \TODO templates

```

```

427 \let\shoveleft\eq@adjust@shoveleft
428 \let\shovecenter\eq@adjust@shovecenter
429 \let\shoveright\eq@adjust@shoveright
430 \let\eq@box@lastfield\eq@box@lastfield@lines
431 \eq@halign@init{%
432 (dev)\eq@dev{starting line \the\eq@row}%
433 \global\advance\eq@row@\@ne
434 }%
435 \tabskip\z@skip
436 \halign\bgroup
437 &%
438 \eq@shape@pos@\m@ne
439 \setbox\eq@fieldbox@\hbox{%
440 \eq@strut@field
441 \@lign
442 $\m@th\displaystyle
443 \eq@hook@colin
444 ##%
445 \eq@punct@apply@col
446 \eq@hook@colout
447 \eq@tagging@mathsave
448 $%
449 \eq@tagging@mathaddlast
450 }%
451 \ifnum\eq@shape@pos@=\m@ne
452 \eq@shape@eval
453 \fi
454 \ifcase\eq@shape@pos@
455 \kern\eq@shape@amount@
456 \box\eq@fieldbox@
457 \skip@\@flushglue
458 \advance\skip@\eq@paddingleft@relax
459 \advance\skip@\eq@paddingright@relax
460 \advance\skip@-\eq@shape@amount@relax
461 \hskip\skip@
462 \eq@tagging@alignleft
463 \or
464 \skip@\@flushglue
465 \advance\skip@\eq@paddingleft@relax
466 \hskip\skip@
467 \box\eq@fieldbox@
468 \skip@\@flushglue
469 \advance\skip@\eq@paddingright@relax
470 \hskip\skip@
471 \eq@tagging@aligncenter
472 \or
473 \skip@\@flushglue
474 \advance\skip@\eq@paddingleft@relax
475 \advance\skip@\eq@paddingright@relax
476 \hskip\skip@
477 \box\eq@fieldbox@
478 \eq@tagging@alignright
479 \fi
480 \tabskip\eq@colsep@relax
481 \crrc
482 \noalign{%
483 \eq@hook@blockbefore
484 }%

```

```

485 \eql@hook@blockin
486 }

487 \def\eql@box@close{%
488 \ifvmode\else
489 \global\eql@totalrows@\eql@row@
490 \eql@punct@apply@block
491 \eql@box@cr@[ \z@skip]%
492 \fi
493 \crrr
494 \noalign{%
495 \eql@hook@blockafter
496 }%
497 \eql@tagging@tablesaveinner
498 \egroup
499 }

```

\eql@box@start

```

500 \def\eql@box@start{%
501 \relax
502 \ifmmode
503 \let\eql@box@endmath\@empty
504 \else
505 $\let\eql@box@endmath=$%
506 \fi
507 \eql@nextopt@process{equationsbox}%
508 \let\eql@punct@block\eql@punct@main
509 \let\eql@punct@main\relax
510 \eql@colsep@\glueexpr\eql@box@colsep\relax
511 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
512 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
513 \eql@indent@\glueexpr\eql@indent@val\relax
514 \eql@stack@save@boxed
515 \let\eql@flushleft\eql@false
516 \eql@row@\z@
517 \eql@totalrows@\@M
518 \eql@shape@sel
519 \hskip\glueexpr\eql@box@marginleft\relax
520 \eql@box@box\bgroup
521 \eql@display@leave
522 \let\\eql@box@cr
523 \ifdefined\eql@box@mode@lines
524 \expandafter\eql@box@open@lines
525 \else
526 \expandafter\eql@box@open@align
527 \fi
528 }

```

\eql@box@end

```

529 \newcommand{\eql@box@end}{%
530 \eql@box@close
531 \egroup
532 \eql@tagging@tableaddinner
533 \hskip\glueexpr\eql@box@marginright\relax
534 \eql@stack@restore
535 \eql@box@endmath
536 }

```

equationsbox (*env.*)

```
537 \newenvironment{equationsbox}{%
538 (dev)\eql@dev@enterenv
539 \eql@ampprotect\eql@box@testall\eql@box@start
540 }{%
541 \eql@box@end
542 (dev)\eql@dev@leaveenv
543 }

544 \def\eql@box@testall{\eql@box@testtilde}
545 \def\eql@box@testtilde#1{%
546 \eql@ifnextgobble@tight~%
547 {\eqnadopt{lines}\eql@box@testopt{#1}}%
548 {\eql@box@testopt{#1}}}
549 \def\eql@box@testopt#1{%
550 \eql@ifnextchar@tight[%]
551 {\eql@box@adopt{#1}}%
552 {#1}}
553 \def\eql@box@adopt#1[#2]{\eqnadopt{#2}#1}

554 \def\eql@mode@aligned{\let\eql@box@mode@lines\eql@false}
555 \def\eql@mode@lined{\let\eql@box@mode@lines\eql@true}
556 \eql@mode@aligned
```

H Equation Numbering

TODO: describe

H.1 Tag Formatting

TODO: describe

```
557 \def\eql@tag@setbox@#1{%
558 \def\eql@tag@box##1{#1}%
559 }
560 \def\eql@tag@setbox#1{%
561 \def\eql@tag@box##1{\hbox{\m@th\normalfont#1}}%
562 }
```

TODO: describe

```
563 \def\eql@tag@setform@#1{%
564 \def\eql@tag@form##1{#1}%
565 }
566 \def\eql@tag@setform#1#2#3{%
567 \def\eql@tag@form##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
568 }

569 \eql@tag@setbox{#1}
570 \eql@tag@setform({#1})
571 \def\eql@tag@boxedform#1{\eql@tag@box{\eql@tag@form{#1}}}
```

Raise Tags

settag@amount@ (*dimen*)

```
572 \newdimen\eql@raisetag@amount@
```

`\raisetag`

```
573 \def\eq@raisetag@default{%
574   \eq@warning{\string\raisetag\space not allowed here}
575   \@gobble
576 }
```

TODO: describe

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

TODO: maybe introduce a star form to enforce raise?

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

H.2 Showkeys Integration

TODO: describe

```
579 \let\eq@SK@loaded\eq@false
580 \let\eq@SK@label\@gobble
581 \let\eq@SK@clearlabel\@empty
582 \let\eq@SK@setlabel\@gobble
583 \let\eq@SK@printlabel@right\@empty
584 \let\eq@SK@printlabel@left\@empty
585 \let\eq@SK@printlabel@line\@empty
586 \def\eq@label@clean{\eq@label@org}
587 \AddToHook{package/showkeys/after}{
588   \let\eq@SK@loaded\eq@true
589   \def\eq@SK@label#1{\SK@\SK@@label#1}
590   \def\eq@SK@clearlabel{\let\eq@SK@lab\relax}
591   \eq@SK@clearlabel
592   \def\eq@SK@@label#1>#2\SK@{%
593     \def\eq@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
594   }
595   \def\eq@SK@setlabel#1{\SK@\eq@SK@@label#1}
596   \def\eq@SK@printlabel@right{%
597     \ifx\eq@SK@lab\relax\else
598       \rlap{\kern\marginparsep\eq@SK@lab}%
599       \eq@SK@clearlabel
600     \fi
601   }
602   \def\eq@SK@printlabel@left{%
603     \ifx\eq@SK@lab\relax\else
604       \llap{\eq@SK@lab\kern\marginparsep}%
605       \eq@SK@clearlabel
606     \fi
607   }
608   \def\eq@SK@printlabel@line{%
609     \ifx\eq@SK@lab\relax\else
610       \dimen@prevdepth
611       \nointerlineskip
612       \ifdefined\eq@tagsleft
613         \llap{%
614           \eq@SK@lab
615           \kern\marginparsep
616         }%
617       \eq@SK@clearlabel
618     \else
```

```

619     \rlap{%
620         \dimen@\displaywidth
621         \advance\dimen@\marginparsep
622         \kern\dimen@
623         \eq@SK@lab
624     }%
625     \fi
626     \eq@SK@clearlabel
627     \prevdepth\dimen@
628     \fi
629 }
630 \let\eq@label@org\label
631 \def\eq@label@clean{\let\SK@\@gobbletwo\eq@label@org}
632 }

```

H.3 Labels

TODO: describe

```

633 % \TODO implement (via label[] or labelname similar to label/tag)
634 \let\eq@nextlabel\@undefined
635 \def\eq@labelname@default{[equation]}

```

\eq@label@org

```

636 \let\eq@label@org\label

```

H.4 Tags

TODO: describe

```

637 \let\eq@nexttag\@undefined

```

\eq@tag@default

```

638 \def\eq@tag@default{%
639     \eq@error{\string\tag\space not allowed here}{}\eq@tag@gobble}
640 \let\tag\eq@tag@default

```

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

```

641 \def\eq@tag@gobble@[#1]#2{}
642 \def\eq@tag@gobble{%
643     \eq@ampprotecttwo\eq@teststaropt@tight\eq@tag@gobble@\eq@tag@gobble@{}}

```

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

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

hyperref anchors

```

644 \let\eq@Hy@anchor\@gobble
645 \AddToHook{package/hyperref/after}{
646     \def\eq@Hy@anchor#1{%
647         \Hy@raisedlink{\hyper@anchor{#1}}%
648     }%
649 }

```

```

650 \def\eq@tag@makenext{%
651     \eq@ampprotecttwo\eq@teststaropt@tight

```

```

652 \eql@tag@makenext@star\eql@tag@makenext@\eql@tag@text
653 }

```

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

```

654 \def\eql@tag@makenext@star[#1]#2{%
655 \global\def\eql@nexttag{%
656 \let\eql@tag@tool\@firstofone
657 \protected@edef\eql@tag@text{#2}%
658 \protected@edef\eql@tag@label{#1}%
659 }%
660 }
661 \def\eql@tag@makenext@[#1]#2{%
662 \global\def\eql@nexttag{%
663 \let\eql@tag@tool\eql@tag@form
664 \protected@edef\eql@tag@text{#2}%
665 \protected@edef\eql@tag@label{#1}%
666 \protected@edef\eql@tag@label{\p@equation\eql@tag@label}%
667 }%
668 }

```

H.5 Anchors

TODO: describe

`g@refcount@` (*counter*)

```

669 \newcount\eql@numbering@refcount@
670 \eql@numbering@refcount@\z@
671 \def\eql@numbering@ref{equation.eql-\the\eql@numbering@refcount@}
672 \def\eql@numbering@refstep{\global\advance\eql@numbering@refcount@\@ne}

```

TODO: describe

```

673 \def\eql@numbering@makeblockanchor{%
674 \eql@numbering@refstep
675 \global\edef\eql@label@currentHref{\eql@numbering@ref}%
676 \eql@Hy@anchor\eql@label@currentHref
677 \global\edef\eql@label@thepage{\thepage}%
678 }
679 \def\eql@numbering@setblockanchor{%
680 \let\thepage\eql@label@thepage
681 \let\@currentHref\eql@label@currentHref
682 }

```

H.6 Tag Composition

TODO: describe

```

\eql@compose@anchor
\eql@compose@tag
\eql@compose@label
683 \def\eql@compose@anchor{%
684 \ifdefined\eql@nexttag
685 \eql@nexttag
686 \def\@currentcounter{equation}%
687 \let\@currentlabel\eql@tag@label
688 \eql@numbering@refstep
689 \edef\@currentHref{\eql@numbering@ref}%

```

```

690 \eql@Hy@anchor\@currentHref
691 \global\let\eql@nexttag\@undefined
692 \else
693 \refstepcounter{equation}%
694 \let\eql@tag@tool\eql@tag@form
695 \edef\eql@tag@text{\theequation}%
696 \fi
697 }

698 \def\eql@compose@label{%
699 \ifmeasuring@\else
700 \eql@SK@clearlabel
701 \ifdefined\eql@nextlabel
702 \ifnum
703 \ifnum\eql@numbering@target@<\z@
704 \eql@row@
705 \else
706 \eql@numbering@target@
707 \fi=\eql@row@
708 \eql@compose@label@
709 \fi
710 \fi
711 \fi
712 }

```

TODO: describe

```

713 \def\eql@compose@label@{%
714 \eql@SK@setlabel\eql@nextlabel
715 \begingroup
716 \ifnum\eql@numbering@target@=\eql@row@
717 \eql@numbering@setblockanchor
718 \fi
719 \let\@currentlabelname\eql@labelname@default
720 \expandafter\eql@label@clean\expandafter{\eql@nextlabel}%
721 \global\let\eql@nextlabel\@undefined
722 \endgroup
723 }

```

TODO: describe

```

724 \def\eql@compose@tag{%
725 \eql@tagging@tagbegin
726 \eql@tag@box{%
727 \eql@tag@tool\eql@tag@text
728 \eql@tagging@tagsave
729 }%
730 \eql@tagging@tagend
731 }

```

TODO: describe

```

732 \def\eql@compose@print{%
733 \eql@compose@anchor
734 \eql@compose@label
735 \ifdefined\eql@tagsleft
736 \eql@SK@printlabel@left
737 \eql@compose@tag
738 \else
739 \eql@compose@tag

```

```

740 \eql@SK@printlabel@right
741 \fi
742 }

```

TODO: describe

```

743 \def\eql@compose@measure{%
744 \ifdefined\eql@nexttag
745 \eql@nexttag
746 \eql@tag@box{\eql@tag@tool\eql@tag@text}%
747 \else
748 \stepcounter{equation}%
749 \eql@tag@box{\eql@tag@form\theequation}%
750 \fi
751 \ifnum\eql@numbering@target@<\z@
752 \global\let\eql@nextlabel\@undefined
753 \global\let\eql@nexttag\@undefined
754 \fi
755 }

```

TODO: describe

```

756 \def\eql@compose@null{%
757 \ifdefined\eql@nexttag\else
758 \stepcounter{equation}%
759 \fi
760 \ifnum\eql@numbering@target@<\z@
761 \global\let\eql@nextlabel\@undefined
762 \global\let\eql@nexttag\@undefined
763 \fi
764 }

```

H.7 Tagbox Methods

TODO: describe

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

```

765 \def\eql@tagbox@make#1{%
766 \setbox\eql@tagbox@\hbox{\eql@strut@tag@\@lign#1}%
767 \eql@tagwidth@\wd\eql@tagbox@
768 \ifdim\eql@tagwidth@<\eql@tagwidthmin@
769 \eql@tagwidth@\eql@tagwidthmin@
770 \fi
771 \advance\eql@tagwidth@\eql@tagsepmin@
772 }

```

TODO: describe

```

773 \def\eql@tagbox@print@right{%
774 \kern-\wd\eql@tagbox@
775 \box\eql@tagbox@
776 }

```

TODO: describe

```

777 \def\eql@tagbox@print@left{%
778 \wd\eql@tagbox@\z@
779 \box\eql@tagbox@
780 }

```

TODO: describe

```
781 \def\eq@tagbox@print@right@raise{%
782   \ifnum\eq@row@=\eq@totalrows@
783     \global\advance\eq@raisetag@firstlast@\tw@
784   \fi
785   \llap{\vtop{%
786     \vskip-\eq@raisetag@amount@
787     \normalbaselines
788     \setbox\@ne\hbox{}}%
789     \dp\@ne\eq@line@depth@
790     \box\@ne
791     \box\eq@tagbox@
792   }}%
793 }
794 \def\eq@tagbox@print@left@raise{%
795   \ifnum\eq@row@=\@ne
796     \global\advance\eq@raisetag@firstlast@\@ne
797   \fi
798   \rlap{\vbox{%
799     \normalbaselines
800     \box\eq@tagbox@
801     \vbox to\eq@line@height@{}}%
802     \vskip\eq@raisetag@amount@
803   }}%
804 }
```

TODO: describe

```
805 \def\eq@numbering@printsubeqlabel{%
806   \ifdefined\eq@parentlabel
807     \eq@numbering@makeblockanchor
808     \eq@SK@setlabel\eq@parentlabel
809     \begingroup
810       \def\@currentcounter{equation}%
811       \eq@numbering@setblockanchor
812       \let\@currentlabelname\eq@labelname@default
813       \protected@edef\@currentlabel{\p@equation\theparentequation}%
814       \expandafter\eq@label@clean\expandafter{\eq@parentlabel}%
815     \endgroup
816     \eq@SK@printlabel@line
817   \fi
818 }
```

H.8 Numbering Schemes

TODO: describe

```
819 \def\eq@numbering@tab@first{first}
820 \def\eq@numbering@tab@last{last}
821 \def\eq@numbering@tab@middle{middle}
822 \def\eq@numbering@tab@here{here}
823 \def\eq@numbering@tab@in{in}
824 \def\eq@numbering@tab@out{out}
825 \def\eq@numbering@tab@sub{sub}
826 \def\eq@numbering@tab@all{all}
827 \def\eq@numbering@tab@none{none}
```

TODO: describe

```

828 \let\eq@numbering@tab@f\eq@numbering@tab@first
829 \let\eq@numbering@tab@l\eq@numbering@tab@last
830 \let\eq@numbering@tab@m\eq@numbering@tab@middle
831 \let\eq@numbering@tab@mid\eq@numbering@tab@middle
832 \let\eq@numbering@tab@o\eq@numbering@tab@out
833 \let\eq@numbering@tab@outside\eq@numbering@tab@out
834 \let\eq@numbering@tab@i\eq@numbering@tab@in
835 \let\eq@numbering@tab@inside\eq@numbering@tab@in
836 \let\eq@numbering@tab@within\eq@numbering@tab@in
837 \let\eq@numbering@tab@h\eq@numbering@tab@here
838 \let\eq@numbering@tab@s\eq@numbering@tab@sub
839 \let\eq@numbering@tab@subeq\eq@numbering@tab@sub
840 \let\eq@numbering@tab@subequation\eq@numbering@tab@sub
841 \let\eq@numbering@tab@subequations\eq@numbering@tab@sub
842 \let\eq@numbering@tab@a\eq@numbering@tab@all
843 \let\eq@numbering@tab@n\eq@numbering@tab@none
844 \expandafter\let\csname eq@numbering@tab@!\endcsname\eq@numbering@tab@all
845 \expandafter\let\csname eq@numbering@tab@*\endcsname\eq@numbering@tab@none
846 \expandafter\let\csname eq@numbering@tab@1\endcsname\eq@numbering@tab@first

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

848 \def\eq@numbering@set#1{%
849   \ifcsname eq@numbering@tab@#1\endcsname
850     \expandafter\let\expandafter\eq@numbering@mode
851     \csname eq@numbering@tab@#1\endcsname
852   \else
853     \eq@error{numbering mode ‘#1’ unknown: setting to ‘all’}%
854     \let\eq@numbering@mode\eq@numbering@tab@all
855   \fi
856 }

ing@target@ (counter)

857 \let\eq@numbering@active\eq@true
858 \newcount\eq@numbering@target@

859 \def\eq@numbering@mode@all{%
860   \eq@numbering@target@\m@ne}
861 \def\eq@numbering@mode@sub{%
862   \eq@numbering@target@\m@ne
863   \let\eq@numbering@subeq@use\eq@true}
864 \def\eq@numbering@mode@none{%
865   \eq@numbering@target@\m@ne
866   \let\eq@numbering@active\eq@false}
867 \def\eq@numbering@mode@first{%
868   \eq@numbering@target@\@ne}
869 \def\eq@numbering@mode@last{%
870   \eq@numbering@target@\@MM}
871 \def\eq@numbering@mode@here{%
872   \eq@numbering@target@\z@}

```

TODO: describe

```

873 \def\eq@numbering@mode@in{%
874   \ifdefined\eq@tagsleft
875     \eq@numbering@mode@last
876   \else
877     \eq@numbering@mode@first

```

```
878 \fi
879 }
```

TODO: describe

```
880 \def\eq@numbering@mode@out{%
881 \ifdefined\eq@tagsleft
882 \eq@numbering@mode@first
883 \else
884 \eq@numbering@mode@last
885 \fi
886 }
```

TODO: describe

```
887 \def\eq@numbering@mode@middle{%
888 \eq@numbering@target@z@
889 \let\eq@numbering@eval@target\eq@numbering@eval@middle}
890 \def\eq@numbering@eval@middle{%
891 \ifnum\eq@numbering@target@=\z@
892 \count@\eq@row@
893 \advance\count@\@ne
894 \divide\count@\tw@
895 \global\eq@numbering@target@\count@
896 \fi
897 }
```

TODO: describe

```
898 \def\eq@numbering@eval@mode{%
899 \let\eq@numbering@eval@target@\undefined
900 \let\eq@numbering@subeq@use\eq@false
901 \csname eq@numbering@mode@\eq@numbering@mode\endcsname
902 \ifdefined\eq@numbering@active
903 \let\eq@numbering@eqnswinit\@eqnswtrue
904 \else
905 \let\eq@numbering@eqnswinit\@eqnswfalse
906 \fi
907 \let\eq@numbering@active\eq@false
908 }
```

TODO: reconsider operation

`\numberhere`

```
909 \def\numberhere{%
910 \ifmeasuring@
911 \ifnum\eq@numbering@target@<\z@\else
912 \global\eq@numbering@target@\eq@row@
913 \fi
914 \fi
915 }
```

TODO: describe

`\numbernext`

```
916 \def\numbernext{%
917 \ifmeasuring@
918 \ifnum\eq@numbering@target@<\z@\else
919 \ifnum\eq@numbering@target@=\eq@row@
```

```

920     \global\advance\eql@numbering@target@\@ne
921     \fi
922     \fi
923     \fi
924 }

```

H.9 Numbering Framework

TODO: describe

```

925 \let\eql@numbering@autolabel\eql@false
926 \let\eql@numbering@autotag\eql@true
927 \let\eql@numbering@blocklabel\@undefined
928 \let\eql@numbering@blocktag\@undefined

929 \eql@amsmath@after{
930   \let\eql@print@eqnum@default\print@eqnum
931   \let\eql@incr@eqnum@default\incr@eqnum
932 }

```

TODO: describe

```

933 \def\donumber{%
934   \if@eqnsw\else
935     \global\@eqnswtrue
936     \ifx\print@eqn\@empty
937       \global\let\print@eqn\eql@print@eqnum@default
938     \fi
939     \ifx\incr@eqn\@empty
940       \global\let\incr@eqn\eql@incr@eqnum@default
941     \fi
942   \fi
943 }

```

TODO: describe

```

944 \def\eql@label@warn{%
945   \ifdefined\eql@numbering@autolabel
946     \global\@eqnswtrue
947   \fi
948   \ifdefined\eql@nextlabel
949     \eql@warn@label@multiple\eql@nextlabel
950   \fi
951   \global\edef\eql@nextlabel
952 }

```

TODO: describe

```

953 \def\eql@tag@warn{%
954   \ifdefined\eql@numbering@autotag
955     \global\@eqnswtrue
956   \fi
957   \ifdefined\eql@nexttag
958     \eql@warn@tag@multiple
959   \fi
960   \eql@tag@makenext
961 }

```

TODO: describe

```

962 \def\eqlabel@nowarn{%
963   \ifdefined\eq@numbering@autolabel
964     \global\@eqnswtrue
965   \fi
966   \global\edef\eq@nextlabel
967 }

```

TODO: describe

```

968 \def\eq@tag@nowarn{%
969   \ifdefined\eq@numbering@autotag
970     \global\@eqnswtrue
971   \fi
972   \eq@tag@makenext
973 }

```

TODO: describe

```

974 \def\eq@blocklabel@set#1{%
975   \ifdefined\eq@blocklabel
976     \eq@warn@label@multiple\eq@blocklabel
977   \fi
978   \edef\eq@blocklabel{#1}%
979 }

```

TODO: describe

```

980 \def\eq@blocktag@set#1{%
981   \ifdefined\eq@blocktag
982     \eq@warn@tag@multiple
983   \fi
984   \def\eq@blocktag{{#1}}%
985 }

```

TODO: describe

```

986 \def\eq@blocktag@setstar#1{%
987   \ifdefined\eq@blocktag
988     \eq@warn@tag@multiple
989   \fi
990   \def\eq@blocktag{*#1}%
991 }

```

Single-Line TODO: describe

```

992 \def\eq@numbering@single@init{%
993   \let\label\eqlabel@warn
994   \let\tag\eq@tag@warn
995   \let\raisetag\eq@raisetag
996   \eq@numbering@target@m@ne
997   \let\eq@nextlabel\eq@blocklabel
998   \ifdefined\eq@blocktag
999     \expandafter\eq@tag@makenext\eq@blocktag
1000  \else
1001    \let\eq@nexttag\@undefined
1002  \fi
1003  \eq@numbering@eqnswinit
1004  \ifdefined\eq@numbering@autolabel
1005    \ifdefined\eq@nextlabel
1006      \@eqnswtrue

```

```

1007   \fi
1008 \fi
1009 \ifdefined\eql@numbering@autotag
1010   \ifdefined\eql@nexttag
1011     \@eqnswtrue
1012   \fi
1013 \fi
1014 \global\eql@raisetag@amount@ \z@
1015 }

```

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

```

1016 \def\eql@numbering@measure@init{%
1017   \let\label\eql@label@warn
1018   \let\tag\eql@tag@warn
1019   \let\raisetag\eql@raisetag
1020   \global\let\eql@nextlabel\eql@blocklabel
1021   \ifdefined\eql@blocktag
1022     \expandafter\eql@tag@makenext\eql@blocktag
1023   \else
1024     \global\let\eql@nexttag@undefined
1025   \fi
1026   \ifnum\eql@numbering@target@<\z@\else
1027     \eql@numbering@eqnswinit
1028     \ifdefined\eql@numbering@autolabel
1029       \ifdefined\eql@nextlabel
1030         \@eqnswtrue
1031       \fi
1032     \fi
1033 \fi
1034 }

```

TODO: describe

```

1035 \def\eql@numbering@measure@line@begin{%
1036   \ifnum\eql@numbering@target@<\z@
1037     \global\eql@numbering@eqnswinit
1038   \fi
1039 }

```

TODO: describe

```

1040 \def\eql@numbering@measure@eval{%
1041   \ifdefined\eql@numbering@eval@target
1042     \eql@numbering@eval@target
1043   \fi
1044   \ifnum\eql@numbering@target@>\eql@row@
1045     \global\eql@numbering@target@ \eql@row@
1046   \fi
1047   \ifnum\eql@numbering@target@>\z@
1048     \if@eqnsw\else
1049       \global\eql@numbering@target@ \z@
1050     \fi
1051   \fi
1052   \ifnum\eql@numbering@target@<\@ne
1053     \ifdefined\eql@nextlabel
1054       \eql@warn@label@unused
1055     \global\let\eql@nextlabel@undefined
1056   \fi

```

```

1057 \ifdefined\eq@nexttag
1058 \eq@warn@tag@unused
1059 \global\let\eq@nexttag\@undefined
1060 \fi
1061 \fi
1062 }

```

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

```

1063 \def\eq@numbering@print@init{%
1064 \ifnum\eq@numbering@target@<\z@
1065 \let\label\eq@label@nowarn
1066 \let\tag\eq@tag@nowarn
1067 \let\raisetag\eq@raisetag
1068 \let\eq@nextlabel\eq@blocklabel
1069 \ifdefined\eq@blocktag
1070 \expandafter\eq@tag@makenext\eq@blocktag
1071 \else
1072 \let\eq@nexttag\@undefined
1073 \fi
1074 \else
1075 \let\label\@gobble
1076 \let\tag\eq@tag@gobble
1077 \let\raisetag\eq@gobble
1078 \fi
1079 }

```

TODO: describe

```

1080 \def\eq@numbering@print@block@begin{%
1081 \ifnum\eq@numbering@target@>\z@
1082 \eq@numbering@makeblockanchor
1083 \fi
1084 \ifdefined\eq@numbering@subeq@use
1085 \eq@numbering@printsubeqlabel
1086 \fi
1087 }

```

TODO: describe

```

1088 \def\eq@numbering@print@line@begin{%
1089 \ifnum\eq@numbering@target@<\z@
1090 \global\eq@numbering@eqnswinit
1091 \global\eq@raisetag@amount@\z@
1092 \fi
1093 }

```

TODO: describe

```

1094 \def\eq@numbering@print@line@eval{%
1095 \ifnum\eq@numbering@target@<\z@\else
1096 \ifnum\eq@numbering@target@=\eq@row@
1097 \global\@eqnswtrue
1098 \else
1099 \global\@eqnswfalse
1100 \fi
1101 \fi
1102 }

```

I Subequation Numbering

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

I.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:

```
1103 \eql@amsmath@undefine\c@parentequation
1104 \eql@amsmath@undefine\theparentequation
1105 \ifdefined\c@parentequation\else
1106 \newcounter{parentequation}
1107 \fi
```

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

```
1108 \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:

```
1109 \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`:

```
1110 \def\eql@subequations@init{%
1111   \edef\eql@subequations@restorecounter{%
1112     \global\c@equation\the\c@equation\relax}%
1113   \ifdefined\eql@blocktag
1114     \expandafter\eql@tag@makenext\eql@blocktag
1115     \eql@nexttag
1116     \eql@numbering@refstep
1117     \protected@edef\theHparentequation{\eql@numbering@ref}%
1118     \protected@edef\theparentequation{\eql@tag@text}%
1119   \else
1120     \advance\c@equation\@ne
1121     \protected@edef\theparentequation{\theequation}%
1122     \ifdefined\theHequation
1123       \protected@edef\theHparentequation{\theHequation}%
1124     \fi
1125   \fi
1126   \global\c@parentequation\c@equation
1127   \global\c@equation\z@
1128   \let\theequation\eql@subequations@template
1129   \def\theHequation{\theHparentequation.\arabic{equation}}%
1130 }
```

`\eql@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:

```

1131 \def\eql@subequations@close{%
1132   \ifnum\c@equation=\z@
1133     \eql@subequations@restorecounter
1134   \else
1135     \global\c@equation\c@parentequation
1136   \fi
1137 }
```

I.2 Environment

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

TODO: join with other similar anchor routines `\eql@numbering@printsubequlabel`

```

1138 \def\eql@subequations@start{%
1139   \let\eql@blocktag\undefined
1140   \let\eql@blocklabel\undefined
1141   \eql@nextopt@process{subequations}%
1142   \eql@subequations@init
1143   \eql@numbering@refstep
1144   \edef\eql@subequations@currentHref{\eql@numbering@ref}%
1145   \eql@Hy@anchor\eql@subequations@currentHref
1146   \edef\eql@subequations@thepage{\thepage}%
1147   \def\@currentcounter{equation}%
1148   \let\@currentHref\eql@subequations@currentHref
1149   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1150   \let\@currentlabelname\eql@labelname@default
1151   \let\eql@subequations@active\eql@true
1152   \ifdefined\eql@blocklabel
1153     \eql@SK@label\eql@blocklabel
1154   \fi
1155   \ignorespaces
1156 }
```

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

```

1157 \def\eql@subequations@end{%
1158   \ifnum\c@equation>\z@
1159     \ifdefined\eql@blocklabel
1160       \begingroup
1161         \def\@currentcounter{equation}%
1162         \let\thepage\eql@subequations@thepage
1163         \let\@currentHref\eql@subequations@currentHref
1164 % \TODO how about tag* ?! also within equations!
1165         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1166         \let\@currentlabelname\eql@labelname@default
1167         \expandafter\eql@label@clean\expandafter{\eql@blocklabel}%
1168       \endgroup
1169     \fi
1170   \fi
1171   \eql@subequations@close
1172   \ignorespacesafterend

```

1173 }

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

```
1174 \newenvironment{eql@subequations}{%
1175 (dev)\eql@dev@enterenv
1176 \eql@subequations@testall\eql@subequations@start%
1177 }{%
1178 \eql@subequations@end
1179 (dev)\eql@dev@leaveenv
1180 }
```

TODO: describe

```
1181 \def\eql@subequations@testall{\eql@subequations@testopt}
1182 \def\eql@subequations@testopt#1{%
1183 \eql@ifnextchar@tight[%]
1184 {\eql@subequations@adopt{\eql@subequations@testat{#1}}}%
1185 {\eql@subequations@testat{#1}}
1186 \def\eql@subequations@adopt#1[#2]{\eqnadopt{#2}#1}
1187 \def\eql@subequations@testat#1{%
1188 \eql@ifat@tight%
1189 {\eql@subequations@addlabel{#1}}%
1190 {#1}}
1191 \def\eql@subequations@addlabel#1#2{\eqnadopt{label={#2}}#1}
```

I.3 Subequation Scheme

TODO: describe

```
1192 \def\eql@numbering@subeq@init{%
1193 \let\eql@save@theequation\theequation
1194 \let\eql@save@theHequation\theHequation
1195 \eql@subequations@init
1196 \let\eql@parentlabel\eql@blocklabel
1197 \let\eql@parenttag\eql@blocktag
1198 \let\eql@blocklabel\@undefined
1199 \let\eql@blocktag\@undefined
1200 }
```

TODO: describe

```
1201 \def\eql@numbering@subeq@test{%
1202 \ifnum\c@equation<\tw@
1203 \let\eql@numbering@subeq@use\@ne
1204 \fi
1205 }
```

TODO: describe

```
1206 \def\eql@numbering@subeq@revert{%
1207 \let\eql@blocklabel\eql@parentlabel
1208 \let\eql@blocktag\eql@parenttag
1209 \let\eql@numbering@subeq@use\eql@false
1210 \let\theequation\eql@save@theequation
1211 \let\theHequation\eql@save@theHequation
1212 \eql@subequations@restorecounter
1213 }
```

TODO: describe

```
1214 % \TODO note must not use setcounter here (when calc is loaded)
1215 \def\eqL@numbering@subeq@close{%
1216   \eqL@subequations@close
1217 }
```

J Display Equations Support

TODO: describe

J.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1218 \interdisplaylinepenalty\@M
```

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

```
1219 \def\eqL@getdsp@pen#1{%
1220   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1221 }
```

TODO: allow a displaybreak before equations

```
1222 \DeclareRobustCommand{\eqL@displaybreak@default}[1][4]{%
1223   \eqL@warning{Invalid use of \string\displaybreak}{}}
1224 \eqL@amsmath@after{\let\eqL@displaybreak@default\displaybreak}
1225 \eqL@amsmath@let\displaybreak\eqL@displaybreak@default
```

```
1226 \newcount\eqL@displaybreak@pen@
1227 \newcount\eqL@displaybreak@prepen@
```

```
1228 \protected\def\eqL@displaybreak@print{%
1229   \eqL@amprotect\eqL@testopt@tight\eqL@displaybreak@print@{4}%
1230 }
```

TODO: describe

```
1231 \def\eqL@displaybreak@print@[#1]{%
1232   \ifnum#1<\z@
1233     \global\eqL@displaybreak@pen@\@MM
1234   \else
1235     \global\eqL@displaybreak@pen@-\@getpen{#1}\relax
1236   \fi
1237 }
```

TODO: describe

```
1238 \def\eqL@displaybreak@pre#1{%
1239   \ifnum#1<\z@
1240     \global\eqL@displaybreak@prepen@\@MM
1241   \else
1242     \global\eqL@displaybreak@prepen@-\@getpen{#1}\relax
1243   \fi
1244 }
```

TODO: describe

```
1245 \protected\def\eqldisplaybreak@measure{%
1246   \eql@ampprotect\eql@testopt@tight\eqldisplaybreak@measure@{4}%
1247 }
1248 \def\eqldisplaybreak@measure#[#1]{}
```

J.2 General Initialisation

TODO: describe

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

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

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

```
1249 \newskip\eql@vspaceskip@
1250 \newskip\eql@abovespace@
1251 \newskip\eql@belowspace@
1252 \let\eql@vspace@org\vspace
1253 \def\eql@vspace{\eql@ifstar@loose\eql@vspace@\eql@vspace@}
1254 \def\eql@vspace@#1{%
1255   \global\advance\eql@vspaceskip@\glueexpr#1\relax}
```

`\eqldisplay@init`

```
1256 \def\eqldisplay@init{%
1257   \eqldisplay@firstavail@<\z@
1258   \eql@raisetag@firstlast@<\z@
1259   \let\displaybreak\eqldisplaybreak@print
1260   \eqldisplaybreak@open@\@MM
1261   \eql@vspaceskip@<\z@skip
1262   \let\eql@vspace@org\vspace
1263   \let\vspace\eql@vspace
1264 }
```

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

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

```
1265 \def\eqldisplay@close{%
1266   \ifdim\eqldisplay@firstavail@<\z@
1267     \eqldisplay@firstavail@<\z@
1268     \fi
1269     \eql@skip@mode@leave@<\z@
1270     \ifdim\eql@halign@prevdepth@=\maxdimen
1271       \ifdim\predisplaysize=-\maxdimen
1272         \eql@skip@mode@above@\tw@
1273         \eql@skip@mode@below@\eql@skip@mode@cont@
1274       \else
1275         \eql@skip@mode@above@\z@
1276         \eql@skip@mode@below@\z@
1277         \advance\eqldisplay@firstavail@\displayindent
1278         \ifdim\eqldisplay@firstavail@>\predisplaysize
1279 % \TODO: process this here or in below ifcase for short?
1280         \ifcase\eql@skip@mode@short@
1281           \or
1282             \eql@skip@mode@above@\@ne
1283           \or
1284             \eql@skip@mode@above@\@ne
1285           \ifnum\eql@row@=\tw@
1286             \eql@skip@mode@below@\@ne
```

```

1287         \fi
1288     \or
1289         \eql@skip@mode@above@\@ne
1290         \eql@skip@mode@below@\@ne
1291     \fi
1292 \fi
1293 \fi
1294 \else
1295     \ifdim\eql@halign@prevdepth@=-\@m\p@
1296         \eql@skip@mode@above@4\relax
1297         \eql@skip@mode@below@\eql@skip@mode@top@
1298         \eql@skip@mode@leave@z@
1299         \predisplaypenaltyz@
1300         \ifcase\eql@skip@mode@top@
1301             \or
1302             \or
1303             \or
1304             \eql@skip@mode@leave@\@ne
1305             \or
1306             \postdisplaypenaltyz@
1307             \eql@skip@mode@leave@tw@
1308         \fi
1309     \else
1310         \eql@skip@mode@above@thr@@
1311         \eql@skip@mode@below@\eql@skip@mode@par@
1312         \eql@skip@mode@leave@z@
1313         \predisplaypenaltyz@
1314         \ifnum\eql@skip@mode@par@=thr@@
1315             \eql@skip@mode@leave@\@ne
1316         \fi
1317     \fi
1318 \fi
1319 \ifdefined\eql@skip@force@above
1320     \eql@skip@mode@above@\eql@skip@force@above\relax
1321 \fi
1322 \ifdefined\eql@skip@force@below
1323     \eql@skip@mode@below@\eql@skip@force@below\relax
1324 \fi
1325 \ifdefined\eql@skip@force@leave
1326     \eql@skip@mode@leave@\eql@skip@force@leave\relax
1327 \fi
1328 \ifodd\eql@raisetag@firstlast@
1329     \ifcase\eql@skip@mode@above@
1330         \abovedisplayskip\glueexpr\eql@skip@tag@above\relax
1331     \or
1332         \abovedisplayskip\glueexpr\eql@skip@tag@above\relax
1333     \or
1334         \abovedisplayskip\glueexpr\eql@skip@tag@above\relax
1335     \or
1336         \abovedisplayskip\glueexpr\eql@skip@partag@above\relax
1337     \or
1338         \abovedisplayskip\glueexpr\eql@skip@partag@above\relax
1339     \or
1340         \abovedisplayskipz@skip
1341     \or
1342         \abovedisplayskip\glueexpr\eql@skip@medtag@above\relax
1343     \or
1344         \abovedisplayskip\glueexpr\eql@skip@custom@above\relax

```

```

1345 \fi
1346 \else
1347 \ifcase\eq\@skip\@mode\@above\@
1348 \abovedisplayskip\glueexpr\eq\@skip\@long\@above\relax
1349 \or
1350 \abovedisplayskip\glueexpr\eq\@skip\@short\@above\relax
1351 \or
1352 \abovedisplayskip\glueexpr\eq\@skip\@cont\@above\relax
1353 \or
1354 \abovedisplayskip\glueexpr\eq\@skip\@par\@above\relax
1355 \or
1356 \abovedisplayskip\glueexpr\eq\@skip\@top\@above\relax
1357 \or
1358 \abovedisplayskip\z\@skip
1359 \or
1360 \abovedisplayskip\glueexpr\eq\@skip\@medtag\@above\relax
1361 \or
1362 \abovedisplayskip\glueexpr\eq\@skip\@custom\@above\relax
1363 \fi
1364 \fi
1365 \ifnum\eq\@raisetag\@firstlast\@>\@ne
1366 \ifcase\eq\@skip\@mode\@below\@
1367 \belowdisplayskip\glueexpr\eq\@skip\@tag\@below\relax
1368 \or
1369 \belowdisplayskip\glueexpr\eq\@skip\@tag\@below\relax
1370 \or
1371 \belowdisplayskip\glueexpr\eq\@skip\@tag\@below\relax
1372 \or
1373 \belowdisplayskip\glueexpr\eq\@skip\@partag\@below\relax
1374 \or
1375 \belowdisplayskip\glueexpr\eq\@skip\@partag\@below\relax
1376 \or
1377 \belowdisplayskip\z\@skip
1378 \or
1379 \belowdisplayskip\glueexpr\eq\@skip\@medtag\@below\relax
1380 \or
1381 \belowdisplayskip\glueexpr\eq\@skip\@custom\@below\relax
1382 \fi
1383 \else
1384 \ifcase\eq\@skip\@mode\@below\@
1385 \belowdisplayskip\glueexpr\eq\@skip\@long\@below\relax
1386 \or
1387 \belowdisplayskip\glueexpr\eq\@skip\@short\@below\relax
1388 \or
1389 \belowdisplayskip\glueexpr\eq\@skip\@cont\@below\relax
1390 \or
1391 \belowdisplayskip\glueexpr\eq\@skip\@par\@below\relax
1392 \or
1393 \belowdisplayskip\glueexpr\eq\@skip\@top\@below\relax
1394 \or
1395 \belowdisplayskip\z\@skip
1396 \or
1397 \belowdisplayskip\glueexpr\eq\@skip\@medtag\@below\relax
1398 \or
1399 \belowdisplayskip\glueexpr\eq\@skip\@custom\@below\relax
1400 \fi
1401 \fi
1402 \ifnum\eq\@displaybreak\@open\@=\@MM\else

```

```

1403 \postdisplaypenalty\eqldisplaybreak@open@
1404 \fi
1405 \ifnum\eqldisplaybreak@prepen@=\@MM\else
1406 \predisplaypenalty\eqldisplaybreak@prepen@
1407 \fi
1408 \advance\abovedisplayskip\eql@abovespace@
1409 \advance\belowdisplayskip\eql@belowspace@
1410 \advance\belowdisplayskip\eql@vspaceskip@
1411 \abovedisplayshortskip\abovedisplayskip
1412 \belowdisplayshortskip\belowdisplayskip
1413 \count@\prevgraf
1414 \advance\count@\eql@row@
1415 \advance\count@-\tw@
1416 \prevgraf\count@
1417 \global\eql@skip@mode@leave@\eql@skip@mode@leave@
1418 }

1419 \def\eqldisplay@leave{%
1420 \let\label\eql@label@org
1421 \let\tag\eql@tag@default
1422 \let\raisetag\eql@raisetag@default
1423 \let\displaybreak\eqldisplaybreak@default
1424 \let\intertext\eql@intertext@default
1425 \let\vspace\eql@vspace@org
1426 }
1427 \expandafter\def\expandafter\@arrayparboxrestore\expandafter{%
1428 \@arrayparboxrestore
1429 \eqldisplay@leave
1430 \ifdefined\eql@ampproof@active
1431 \eql@amprevert
1432 \fi
1433 \@displayfalse
1434 }

```

J.3 halign Support

TODO: describe

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

```

1435 \newbox\eql@strutbox@
1436 \def\eql@strut{\copy\eql@strutbox@}
1437 \let\eql@strut@field\eql@strut
1438 \let\eql@strut@tag\eql@strut
1439 \def\eql@strut@make{%
1440 \setbox\eql@strutbox@\hbox{%
1441 \@tempdima\normalbaselineskip
1442 \advance\@tempdima-\normallineskiplimit
1443 \@tempdimb.3\normalbaselineskip
1444 \advance\@tempdimb.5\normallineskiplimit
1445 \advance\@tempdima-\@tempdimb
1446 \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1447 }
1448 }
1449 \AtBeginDocument{\eql@strut@make}

```

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

```

1450 \def\eq\halign@spread{%
1451   \eq\spread@amount@\glueexpr\eq\spread\relax
1452   \advance\eq\spread@amount@\normalbaselineskip
1453   \advance\eq\spread@amount@-\baselineskip
1454   \ifdim\eq\spread@amount@>\z@
1455     \openup\eq\spread@amount@
1456     \ifdefined\spread@equation
1457       \let\spread@equation\@empty
1458     \fi
1459   \fi
1460 }

```

gn@prevdepth@ (*dimen*)

```

1461 \newdimen\eq\halign@prevdepth@
1462 \def\eq\halign@catchprevdepth{%
1463   \ifvmode
1464     \eq\halign@prevdepth@\prevdepth
1465     \nointerlineskip
1466     \noindent
1467   \else
1468     \eq\halign@prevdepth@\maxdimen
1469   \fi
1470 }

```

```

1471 \def\eq\halign@leave{%
1472   \ifcase\eq\skip@mode@leave@
1473   \or
1474     \endgraf
1475   \or
1476     \endgraf
1477   \prevdepth-\@m\p@
1478   \fi
1479 }

```

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

```

1480 \def\eq\halign@before{%
1481   \ifdim\eq\halign@prevdepth@=\maxdimen\else
1482     \prevdepth\eq\halign@prevdepth@
1483   \fi
1484   \ifdim\prevdepth=-\@m\p@\else
1485     \ifdefined\eq\display@height
1486       \skip@\baselineskip
1487       \advance\skip@-\glueexpr\eq\display@height\relax
1488       \advance\skip@-\prevdepth\relax
1489       \ifdim\skip@<\lineskiplimit
1490         \skip@\lineskip
1491       \fi
1492       \advance\skip@-\eq\spread@amount@\relax
1493       \vskip\skip@
1494       \nointerlineskip
1495     \else
1496       \vskip-\eq\spread@amount@\relax
1497     \fi
1498   \fi
1499 }

```

TODO: describe

```

1500 \def\eq@halign@after{%
1501   \ifdefined\eq@display@depth
1502     \prevdepth\glueexpr\eq@display@depth\relax
1503   \fi
1504 }

```

TODO: describe

```

1505 \def\eq@halign@init#1{%
1506   \eq@halign@spread
1507   \eq@strut@make
1508   \everycr{\noalign{#1}}%
1509 }

```

J.4 Stack

TODO: describe

```

1510 \def\eq@stack@enable{%
1511   \let\eq@stack@save@single\eq@stack@save@single@
1512   \let\eq@stack@save@multi\eq@stack@save@multi@
1513   \let\eq@stack@save@boxed\eq@stack@save@boxed@
1514 }

```

TODO: describe

```

1515 \let\eq@stack@save@single\eq@stack@enable
1516 \let\eq@stack@save@multi\eq@stack@enable
1517 \let\eq@stack@save@boxed\eq@stack@enable
1518 \let\eq@stack@restore\@empty

```

TODO: describe

```

1519 \def\eq@stack@save@reg#1{\global#1\the#1\relax}
1520 \def\eq@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

TODO: describe

```

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

```

TODO: describe

```

1537 \def\eq@stack@save@multi@{%
1538   \let\eq@stack@nextlabel\eq@nextlabel
1539   \let\eq@stack@nexttag\eq@nexttag
1540   \let\eq@stack@tagwidth@tab\eq@tagwidth@tab

```

```

1541 \let\eqL@stack@fieldlength@tab\eqL@fieldlength@tab
1542 \let\eqL@stack@colwidth@tab\eqL@colwidth@tab
1543 \let\eqL@stack@label@thepage\eqL@label@thepage
1544 \let\eqL@stack@label@currentHref\eqL@label@currentHref
1545 \edef\eqL@stack@restore{%
1546   \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1547   \eqL@stack@save@let\eqL@stack@nextlabel\eqL@nextlabel
1548   \eqL@stack@save@let\eqL@stack@nexttag\eqL@nexttag
1549   \eqL@stack@save@let\eqL@stack@tagwidth@tab\eqL@tagwidth@tab
1550   \eqL@stack@save@let\eqL@stack@fieldlength@tab\eqL@fieldlength@tab
1551   \eqL@stack@save@let\eqL@stack@colwidth@tab\eqL@colwidth@tab
1552   \eqL@stack@save@let\eqL@stack@label@thepage\eqL@label@thepage
1553   \eqL@stack@save@let\eqL@stack@label@currentHref\eqL@label@currentHref
1554   \eqL@stack@save@reg\eqL@displaybreak@pen@
1555   \eqL@stack@save@reg\eqL@vspaceskip@
1556   \eqL@stack@save@reg\eqL@shape@pos@
1557   \eqL@stack@save@reg\eqL@shape@amount@
1558   \eqL@stack@save@reg\eqL@display@firstavail@
1559   \eqL@stack@save@reg\eqL@raisetag@amount@
1560   \eqL@stack@save@reg\eqL@raisetag@firstlast@
1561   \eqL@stack@save@reg\eqL@column@
1562   \eqL@stack@save@reg\eqL@totalcolumns@
1563   \eqL@stack@save@reg\eqL@line@avail@
1564   \eqL@stack@save@reg\eqL@line@pos@
1565   \eqL@stack@save@reg\eqL@line@width@
1566   \eqL@stack@save@reg\eqL@line@depth@
1567   \eqL@stack@save@reg\eqL@line@height@
1568   \eqL@stack@save@reg\eqL@numbering@target@
1569   \eqL@stack@save@reg\eqL@row@
1570 }%
1571 }
1572 \def\eqL@stack@save@boxed{%
1573   \edef\eqL@stack@restore{%
1574     \eqL@stack@save@reg\eqL@row@
1575     \eqL@stack@save@reg\eqL@totalrows@
1576     \eqL@stack@save@reg\eqL@shape@pos@
1577     \eqL@stack@save@reg\eqL@shape@amount@
1578   }%
1579 }

```

K Horizontal Spacing for Lines

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

K.1 Supporting Definitionss

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

```

1580 \ifdefined\inf@bad\else%
1581   \newcount\inf@bad
1582   \inf@bad1000000\relax
1583 \fi

```

`\eqL@restore@hfuzz` We need to change the value of `\hfuzz` temporarily. The method `\eqL@save@hfuzz` stores `\eqL@save@hfuzz`

the value for recovery through `\eql@restore@hfuzz`:

```
1584 \let\eql@restore@hfuzz@empty
1585 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

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

```
1586 \newcount\eql@shape@pos@
1587 \newdimen\eql@shape@amount@
```

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

```
1588 \newdimen\eql@marginleft@
1589 \newdimen\eql@marginright@
```

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

```
1590 \newcount\eql@marginbadness@
1591 \newcount\eql@maxbadness@
1592 \eql@marginbadness@ \inf@bad
1593 \eql@maxbadness@ \inf@bad
```

K.2 Shape Schemes

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

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

```
1594 \def\eql@shape@tab@default{default}
1595 \def\eql@shape@tab@left{left}
1596 \def\eql@shape@tab@center{center}
1597 \def\eql@shape@tab@right{right}
1598 \def\eql@shape@tab@first{first}
1599 \def\eql@shape@tab@hanging{hanging}
1600 \def\eql@shape@tab@steps{steps}
```

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

```
1601 \let\eql@shape@tab@def\eql@shape@tab@default
1602 \let\eql@shape@tab@\eql@shape@tab@default
1603 \let\eql@shape@tab@l\eql@shape@tab@left
1604 \let\eql@shape@tab@c\eql@shape@tab@center
1605 \let\eql@shape@tab@r\eql@shape@tab@right
1606 \let\eql@shape@tab@rc\eql@shape@tab@first
1607 \let\eql@shape@tab@indent\eql@shape@tab@first
1608 \let\eql@shape@tab@hang\eql@shape@tab@hanging
1609 \let\eql@shape@tab@lc\eql@shape@tab@hanging
1610 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
1611 \let\eql@shape@tab@lcr\eql@shape@tab@steps
```

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

```
1612 \let\eql@shape@mode\eql@shape@tab@default
```

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

```

1613 \def\eql@shape@set#1{%
1614   \ifcsname eql@shape@tab@#1\endcsname
1615     \expandafter\let\expandafter\eql@shape@mode
1616     \csname eql@shape@tab@#1\endcsname
1617   \else
1618     \eql@error{shape ‘#1’ unknown: setting to default}%
1619     \let\eql@shape@mode\eql@shape@tab@default
1620   \fi
1621 }

```

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

```

1622 \def\eql@shape@center@left{\eql@shape@pos@z@\eql@shape@amount@z@}
1623 \def\eql@shape@center@center{\eql@shape@pos@one\eql@shape@amount@z@}
1624 \def\eql@shape@center@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
1625 \let\eql@shape@center@default\eql@shape@center@center
1626 \def\eql@shape@left@left{\eql@shape@pos@z@\eql@shape@amount@z@}
1627 \def\eql@shape@left@center{\eql@shape@pos@one\eql@shape@amount@z@}
1628 \def\eql@shape@left@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
1629 \let\eql@shape@left@default\eql@shape@left@left

```

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

```

1630 \def\eql@shape@center@first{%
1631   \eql@shape@pos@z@
1632   \eql@shape@amount@z@
1633   \ifnum\eql@totalrows@>\@ne
1634     \ifnum\eql@row@=\@ne
1635       \eql@shape@amount@\eql@indent@
1636     \fi
1637   \fi
1638 }
1639 \def\eql@shape@left@first{%
1640   \eql@shape@pos@z@
1641   \eql@shape@amount@z@
1642   \ifnum\eql@totalrows@>\@ne
1643     \ifnum\eql@row@=\@ne
1644       \eql@shape@amount@\eql@indent@
1645     \fi
1646   \fi
1647 }

```

The `hanging` scheme implements left alignment with hanging indentation for the first line (unless there is only one line). In centered layout all but the first line are indented while in flush-left layout the first line has negative indentation:

```

1648 \def\eql@shape@center@hanging{%
1649   \eql@shape@pos@z@
1650   \eql@shape@amount@\eql@indent@
1651   \ifnum\eql@totalrows@>\@ne
1652     \ifnum\eql@row@=\@ne
1653       \eql@shape@amount@z@
1654     \fi
1655   \fi
1656 }

```

```

1657 \def\eqL@shape@left@hanging{%
1658   \eqL@shape@pos@z@
1659   \eqL@shape@amount@z@
1660   \ifnum\eqL@totalrows@>\@ne
1661     \ifnum\eqL@row@=\@ne
1662       \eqL@shape@amount@-\eqL@indent@
1663     \fi
1664   \fi
1665 }

```

The `steps` scheme implements singles out the first and last lines which are shifted left and right, respectively. In centered layout the shift operates on the alignment whereas in flush-left layout the shift uses indentation:

```

1666 \def\eqL@shape@center@steps{%
1667   \eqL@shape@amount@z@
1668   \eqL@shape@pos@ne
1669   \ifnum\eqL@totalrows@>\@ne
1670     \ifnum\eqL@row@=\@ne
1671       \eqL@shape@pos@z@
1672     \fi
1673     \ifnum\eqL@row@=\eqL@totalrows@
1674       \eqL@shape@pos@tw@
1675     \fi
1676   \fi
1677 }
1678 \def\eqL@shape@left@steps{%
1679   \eqL@shape@pos@z@
1680   \eqL@shape@amount@z@
1681   \ifnum\eqL@totalrows@>\@ne
1682     \ifnum\eqL@row@=\@ne
1683       \eqL@shape@amount@-\eqL@indent@
1684     \fi
1685     \ifnum\eqL@row@=\eqL@totalrows@
1686       \eqL@shape@amount@\eqL@indent@
1687     \fi
1688   \fi
1689 }

```

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

```

1690 \let\eqL@shape@eval\undefined
1691 \def\eqL@shape@sel{%
1692   \expandafter\let\expandafter\eqL@shape@eval
1693   \csname eqL@shape%
1694     @\ifdefined\eqL@flushleft left\else center\fi
1695     @\eqL@shape@mode\endcsname
1696 }

```

`\eqL@adjust@shoveleft` Adjust the alignment of the current equation line. For left alignment an optional argument `\eqL@adjust@shoveright` specifies the amount of indentation:
`\eqL@adjust@shovecenter`

```

1697 \def\eqL@adjust@shoveleft{%
1698   \global\eqL@shape@pos@z@
1699   \eqL@srbgroup\eqL@ifstar@tight
1700   {\eqL@adjust@shoveleft@[\eqL@indent@]}%
1701   {\eqL@ifnextgobble@tight{!}}%
1702   {\eqL@adjust@shoveleft@[-\eqL@indent@]}%

```

```

1703     {\eql@testopt@tight\eql@adjust@shoveleft@z}%
1704   }%
1705 }
1706 \def\eql@adjust@shoveleft#[1]{%
1707   \eql@sregroup\global\eql@shape@amount@glueexpr#1\relax}
1708 \def\eql@adjust@shovecenter{%
1709   \global\eql@shape@pos@one\global\eql@shape@amount@z}
1710 \def\eql@adjust@shoveright{%
1711   \global\eql@shape@pos@tw\global\eql@shape@amount@z}

```

K.3 Adjustment Methods

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

```

1712 \def\eql@adjust@try#1{%
1713   \setbox\eql@fieldbox@\hbox to\displaywidth{%
1714     \unhbox\eql@fieldbox@\unkern\kern#1}%
1715 }

```

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

```

1716 \def\eql@adjust@print#1#2{%
1717   \eql@restore@hfuzz
1718   \hbox to\displaywidth{%
1719     #1%
1720     \unhbox\eql@fieldbox@\unkern
1721     #2%
1722     \eql@tagging@mathaddlast
1723   }%
1724 }

```

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

```

1725 \def\eql@adjust@print@alignleft#1#2{%
1726   \ifnum\eql@row@=\@one
1727     \global\eql@display@firstavail@#1%
1728     \fi
1729     \eql@adjust@print{\kern#1}{\kern#2}%
1730 }
1731 \def\eql@adjust@print@alignright#1#2{%
1732   \ifnum\eql@row@=\@one
1733     \eql@display@firstavail@\displaywidth
1734     \advance\eql@display@firstavail@-\eql@fieldwidth@
1735     \global\advance\eql@display@firstavail@-#2%
1736     \fi
1737     \eql@adjust@print{\kern#1\hfil}{\unskip\kern#2}%
1738 }
1739 \def\eql@adjust@print@aligncenter#1#2{%
1740   \ifnum\eql@row@=\@one
1741     \eql@display@firstavail@\displaywidth
1742     \advance\eql@display@firstavail@-\eql@fieldwidth@
1743     \advance\eql@display@firstavail@#1%
1744     \advance\eql@display@firstavail@-#2%

```

```

1745 \global\divide\eqldisplay@firstavail@tw@
1746 \fi
1747 \eql@adjust@print{\kern#1\hfil}{\kern#2}%
1748 }

```

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

```

1749 \def\eql@adjust@init{%
1750 \eql@save@hfuzz
1751 \hfuzz\maxdimen
1752 \eql@shape@sel
1753 }

```

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

```

1754 \def\eql@adjust@sel@tag{%
1755 \eql@tagging@tagaddbox
1756 \ifcase\eql@shape@pos@
1757 \eql@tagging@alignleft
1758 \or
1759 \eql@tagging@aligncenter
1760 \or
1761 \eql@tagging@alignright
1762 \fi
1763 \csname eql@adjust%
1764 @\ifdefined\eql@flushleft flushleft\else center\fi
1765 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1766 @\ifcase\eql@shape@pos@ shoveleft\or shovecenter\or shoveright\fi
1767 @tag\endcsname
1768 }
1769 \def\eql@adjust@sel@notag{%
1770 \eql@tagging@tagaddbox
1771 \ifcase\eql@shape@pos@
1772 \eql@tagging@alignleft
1773 \or
1774 \eql@tagging@aligncenter
1775 \or
1776 \eql@tagging@alignright
1777 \fi
1778 \csname eql@adjust%
1779 @\ifdefined\eql@flushleft flushleft\else center\fi
1780 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1781 @\ifcase\eql@shape@pos@ shoveleft\or shovecenter\or shoveright\fi
1782 @notag\endcsname
1783 }

```

K.4 Centered Layout

TODO: describe

TODO: check all these!!

```

1784 \def\eql@adjust@center@tagsright@shovecenter@notag{%
1785 \dimen@displaywidth
1786 \advance\dimen@-\eql@fieldwidth@

```

```

1787 \ifdim\dimen@>\eql@tagmargin@
1788   \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1789 \else
1790   \eql@adjust@print@alignleft\z@\z@
1791 \fi
1792 }

```

TODO: describe

```

1793 \def\eql@adjust@center@tagsright@shovecenter@tag{%
1794   \dimen@\displaywidth
1795   \ifdim\eql@tagwidth@<\eql@tagmargin@
1796     \advance\dimen@-\eql@tagmargin@
1797   \else
1798     \advance\dimen@-2\eql@tagwidth@
1799     \advance\dimen@\eql@tagmargin@
1800   \fi
1801   \ifdim\eql@fieldwidth@<\dimen@
1802     \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1803     \eql@tagbox@print@right
1804   \else
1805     \eql@adjust@try\eql@tagwidth@
1806     \ifnum\badness<\eql@maxbadness@
1807       \ifdim\eql@tagwidth@<\eql@tagmargin@
1808         \eql@adjust@print@alignleft\z@\eql@tagwidth@
1809       \else
1810         \eql@adjust@print@alignright\z@\eql@tagwidth@
1811       \fi
1812     \eql@tagbox@print@right
1813   \else
1814     \eql@adjust@center@tagsright@shovecenter@notag
1815     \eql@tagbox@print@right@raise
1816   \fi
1817 \fi
1818 }

```

TODO: describe

```

1819 \def\eql@adjust@center@tagsleft@shovecenter@notag{%
1820   \dimen@\displaywidth
1821   \advance\dimen@-\eql@tagmargin@
1822   \ifdim\eql@fieldwidth@<\dimen@
1823     \eql@adjust@print@aligncenter\eql@tagmargin@\z@
1824   \else
1825     \eql@adjust@print@alignright\z@\z@
1826   \fi
1827 }

```

TODO: describe

```

1828 \def\eql@adjust@center@tagsleft@shovecenter@tag{%
1829   \dimen@\displaywidth
1830   \ifdim\eql@tagwidth@<\eql@tagmargin@
1831     \advance\dimen@-\eql@tagmargin@
1832   \else
1833     \advance\dimen@-2\eql@tagwidth@
1834     \advance\dimen@\eql@tagmargin@
1835   \fi
1836   \ifdim\eql@fieldwidth@<\dimen@
1837     \eql@tagbox@print@left
1838     \eql@adjust@print@aligncenter\eql@tagmargin@\z@

```

```

1839 \else
1840   \eql@adjust@try\eql@tagwidth@
1841   \ifnum\badness<\eql@maxbadness@
1842     \eql@tagbox@print@left
1843     \ifdim\eql@tagwidth@<\eql@tagmargin@
1844       \eql@adjust@print@alignright\eql@tagwidth@z@
1845     \else
1846       \eql@adjust@print@alignleft\eql@tagwidth@z@
1847     \fi
1848   \else
1849     \eql@tagbox@print@left@raise
1850     \eql@adjust@center@tagsleft@shovecenter@notag
1851   \fi
1852 \fi
1853 \eql@display@firstavail@set@z@
1854 }

```

TODO: describe

```

1855 \def\eql@adjust@center@tagsright@shoveleft@notag{%
1856   \dimen@displaywidth
1857   \advance\dimen@-\eql@marginleft@
1858   \advance\dimen@-\eql@shape@amount@
1859   \ifdim\eql@fieldwidth@<\dimen@
1860     \dimen@\eql@marginleft@
1861     \advance\dimen@\eql@shape@amount@
1862     \eql@adjust@print@alignleft\dimen@z@
1863   \else
1864     \eql@adjust@print@alignrightz@z@
1865   \fi
1866 }

```

TODO: describe

```

1867 \def\eql@adjust@center@tagsright@shoveleft@tag{%
1868   \dimen@\eql@marginleft@
1869   \advance\dimen@\eql@shape@amount@
1870   \advance\dimen@\eql@tagwidth@
1871   \eql@adjust@try\dimen@
1872   \ifnum\badness<\eql@marginbadness@
1873     \dimen@\eql@marginleft@
1874     \advance\dimen@\eql@shape@amount@
1875     \eql@adjust@print@alignleft\dimen@\eql@tagwidth@
1876     \eql@tagbox@print@right
1877   \else
1878     \ifdim\eql@marginleft@>-\eql@shape@amount@
1879       \eql@adjust@try\eql@tagwidth@
1880     \fi
1881     \ifnum\badness<\eql@maxbadness@
1882       \eql@adjust@print@alignrightz@\eql@tagwidth@
1883       \eql@tagbox@print@right
1884     \else
1885       \eql@adjust@center@tagsright@shoveleft@notag
1886       \eql@tagbox@print@right@raise
1887     \fi
1888   \fi
1889 }

```

TODO: describe

```

1890 \def\eql@adjust@center@tagsleft@shoveright@notag{%

```

```

1891 \dimen@displaywidth
1892 \advance\dimen@-\eql@tagmargin@
1893 \advance\dimen@-\eql@marginright@
1894 \ifdim\eql@fieldwidth@<\dimen@
1895   \eql@adjust@print@alignright\z@\eql@marginright@
1896 \else
1897   \eql@adjust@print@alignleft\z@\z@
1898 \fi
1899 }

```

TODO: describe

```

1900 \def\eql@adjust@center@tagsleft@shoveright@tag{%
1901   \dimen@\eql@marginright@
1902   \advance\dimen@\eql@tagwidth@
1903   \eql@adjust@try\dimen@
1904   \ifnum\badness<\eql@marginbadness@
1905     \eql@tagbox@print@left
1906     \eql@adjust@print@alignright\eql@tagwidth@\eql@marginright@
1907   \else
1908     \ifdim\eql@marginright@>\z@
1909       \eql@adjust@try\eql@tagwidth@
1910     \fi
1911     \ifnum\badness<\eql@maxbadness@
1912       \eql@tagbox@print@left
1913       \eql@adjust@print@alignleft\eql@tagwidth@\z@
1914     \else
1915       \eql@tagbox@print@left@raise
1916       \eql@adjust@center@tagsleft@shoveright@notag
1917     \fi
1918   \fi
1919   \eql@display@firstavail@set\z@
1920 }

```

TODO: describe

```

1921 \def\eql@adjust@center@tagsright@shoveright@notag{%
1922   \dimen@displaywidth
1923   \advance\dimen@-\eql@tagmargin@
1924   \advance\dimen@-\eql@marginright@
1925   \ifdim\eql@fieldwidth@<\dimen@
1926     \dimen@\eql@tagmargin@
1927     \advance\dimen@\eql@marginright@
1928     \eql@adjust@print@alignright\z@\dimen@
1929   \else
1930     \eql@adjust@print@alignleft\z@\z@
1931   \fi
1932 }

```

TODO: describe

```

1933 \def\eql@adjust@center@tagsright@shoveright@tag{%
1934   \dimen@\eql@tagmargin@
1935   \advance\dimen@\eql@marginright@
1936   \ifdim\eql@tagwidth@<\dimen@
1937     \eql@adjust@try\dimen@%
1938     \ifnum\badness<\eql@marginbadness@
1939       \eql@adjust@print@alignright\z@\dimen@
1940       \eql@tagbox@print@right
1941     \else
1942       \eql@adjust@try\eql@tagwidth@

```

```

1943     \ifnum\badness<\eql@maxbadness@
1944     \eql@adjust@print@alignleft\z@\eql@tagwidth@
1945     \eql@tagbox@print@right
1946     \else
1947     \eql@adjust@print@alignleft\z@\z@
1948     \eql@tagbox@print@left@raise
1949     \fi
1950 \fi
1951 \else
1952 \eql@adjust@try\eql@tagwidth@
1953 \ifnum\badness<\eql@maxbadness@
1954 \eql@adjust@print@alignright\z@\eql@tagwidth@
1955 \eql@tagbox@print@right
1956 \else
1957 \eql@adjust@center@tagsright@shoveright@notag
1958 \eql@tagbox@print@right@raise
1959 \fi
1960 \fi
1961 }

```

TODO: describe

```

1962 \def\eql@adjust@center@tagsleft@shoveleft@notag{%
1963 \dimen@ \displaywidth
1964 \advance\dimen@-\eql@tagmargin@
1965 \advance\dimen@-\eql@marginleft@
1966 \advance\dimen@-\eql@shape@amount@
1967 \ifdim\eql@fieldwidth@<\dimen@
1968 \dimen@ \eql@tagmargin@
1969 \advance\dimen@ \eql@marginleft@
1970 \advance\dimen@ \eql@shape@amount@
1971 \eql@adjust@print@alignleft\dimen@\z@
1972 \else
1973 \eql@adjust@print@alignright\z@\z@
1974 \fi
1975 }

```

TODO: describe

```

1976 \def\eql@adjust@center@tagsleft@shoveleft@tag{%
1977 \dimen@ \eql@tagmargin@
1978 \advance\dimen@ \eql@marginleft@
1979 \advance\dimen@ \eql@shape@amount@
1980 \ifdim\eql@tagwidth@<\dimen@
1981 \eql@adjust@try\dimen@%
1982 \ifnum\badness<\eql@marginbadness@
1983 \eql@tagbox@print@left
1984 \eql@adjust@print@alignleft\dimen@\z@
1985 \else
1986 \eql@adjust@try\eql@tagwidth@
1987 \ifnum\badness<\eql@maxbadness@
1988 \eql@tagbox@print@left
1989 \eql@adjust@print@alignright\eql@tagwidth@\z@
1990 \else
1991 \eql@tagbox@print@left@raise
1992 \eql@adjust@print@alignright\z@\z@
1993 \fi
1994 \fi
1995 \else
1996 \eql@adjust@try\eql@tagwidth@

```

```

1997 \ifnum\badness<\eql@maxbadness@
1998 \eql@tagbox@print@left
1999 \eql@adjust@print@alignleft\eql@tagwidth@\z@
2000 \else
2001 \eql@tagbox@print@left@raise
2002 \eql@adjust@center@tagsleft@shoveleft@notag
2003 \fi
2004 \fi
2005 \eql@display@firstavail@set\z@
2006 }

```

K.5 Flush-Left Layout

TODO: describe

```

2007 \def\eql@adjust@flushleft@shoveleft{%
2008 \eql@marginleft@\eql@flushleftmargin@
2009 \advance\eql@marginleft@\eql@shape@amount@
2010 \ifdim\eql@marginleft@<\eql@flushleftmarginmin@
2011 \eql@marginleft@\eql@flushleftmarginmin@
2012 \fi
2013 \ifdim\eql@marginleft@>\eql@flushleftmarginmax@
2014 \eql@marginleft@\eql@flushleftmarginmax@
2015 \fi
2016 }

```

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

```

2017 \def\eql@adjust@flushleft@shoveleft@notag{%
2018 \ifdim\eql@flushleftmarginmin@<\eql@marginleft@
2019 \eql@adjust@try\eql@marginleft@
2020 \ifnum\badness<\eql@marginbadness@
2021 \eql@adjust@print@alignleft\eql@marginleft@\z@
2022 \else
2023 \eql@adjust@print@alignleft\eql@flushleftmarginmin@\z@
2024 \fi
2025 \else
2026 \eql@adjust@print@alignleft\eql@marginleft@\z@
2027 \fi
2028 }

```

TODO: describe

```

2029 \def\eql@adjust@flushleft@tagsright@shoveleft@notag{%
2030 \eql@adjust@flushleft@shoveleft
2031 \eql@adjust@flushleft@shoveleft@notag
2032 }
2033 \let\eql@adjust@flushleft@tagsleft@shoveleft@notag
2034 \eql@adjust@flushleft@tagsright@shoveleft@notag

```

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

```

2035 \def\eql@adjust@flushleft@tagsright@shoveleft@tag{%
2036 \eql@adjust@flushleft@shoveleft
2037 \dimen@\eql@marginleft@

```

```

2038 \advance\dimen@eql@tagwidth@
2039 \eql@adjust@try\dimen@
2040 \ifnum\badness<\eql@marginbadness@
2041 \eql@adjust@print@alignleft\eql@marginleft@\eql@tagwidth@
2042 \eql@tagbox@print@right
2043 \else
2044 \ifdim\eql@flushleftmarginmin@<\eql@marginleft@
2045 \dimen@\eql@flushleftmarginmin@
2046 \advance\dimen@\eql@tagwidth@
2047 \eql@adjust@try\dimen@
2048 \fi
2049 \ifnum\badness<\eql@maxbadness@
2050 \eql@adjust@print@alignleft\eql@flushleftmarginmin@\eql@tagwidth@
2051 \eql@tagbox@print@right
2052 \else
2053 \eql@adjust@flushleft@shoveleft@notag
2054 \eql@tagbox@print@right@raise
2055 \fi
2056 \fi
2057 }

2058 \def\eql@adjust@flushleft@tagsleft@shoveleft@tag{%
2059 \eql@adjust@flushleft@shoveleft
2060 \ifdim\eql@tagwidth@<\eql@flushleftmarginmin@
2061 \eql@tagbox@print@left
2062 \eql@adjust@flushleft@notag
2063 \else
2064 \ifdim\eql@tagwidth@<\eql@marginleft@
2065 \eql@adjust@try\eql@marginleft@
2066 \ifnum\badness<\eql@marginbadness@
2067 \eql@tagbox@print@left
2068 \eql@adjust@print@alignleft\eql@marginleft@\z@
2069 \else
2070 \eql@adjust@try\eql@tagwidth@
2071 \ifnum\badness<\eql@maxbadness@
2072 \eql@tagbox@print@left
2073 \eql@adjust@print@alignleft\eql@tagwidth@\z@
2074 \else
2075 \eql@tagbox@print@left@raise
2076 \eql@adjust@print@alignleft\eql@flushleftmarginmin@\z@
2077 \fi
2078 \fi
2079 \else
2080 \ifdim\eql@tagwidth@>\eql@flushleftmarginmax@
2081 \eql@tagbox@print@left@raise
2082 \eql@adjust@flushleft@shoveleft@notag
2083 \else
2084 \eql@adjust@try\eql@tagwidth@
2085 \ifnum\badness<\eql@maxbadness@
2086 \eql@tagbox@print@left
2087 \eql@adjust@print@alignleft\eql@tagwidth@\z@
2088 \else
2089 \eql@tagbox@print@left@raise
2090 \eql@adjust@flushleft@shoveleft@notag
2091 \fi
2092 \fi
2093 \fi
2094 \fi
2095 \eql@display@firstavail@set\z@

```

2096 }

TODO: describe

```
2097 \def\eql@adjust@flushleft@shoveright@notag{%
2098   \eql@marginleft@\eql@flushleftmargin@
2099   \ifdim\eql@flushleftmarginmin<\eql@marginleft@
2100     \eql@adjust@try\eql@marginleft@
2101     \ifnum\badness<\eql@marginbadness@
2102       \eql@adjust@print@alignright\eql@marginleft@\z@
2103     \else
2104       \eql@adjust@print@alignright\eql@flushleftmarginmin@\z@
2105     \fi
2106   \else
2107     \eql@adjust@print@alignright\eql@marginleft@\z@
2108   \fi
2109 }
2110 \let\eql@adjust@flushleft@tagsright@shoveright@notag
2111   \eql@adjust@flushleft@shoveright@notag
2112 \let\eql@adjust@flushleft@tagsleft@shoveright@notag
2113   \eql@adjust@flushleft@shoveright@notag
```

TODO: describe

```
2114 \def\eql@adjust@flushleft@tagsright@shoveright@tag{%
2115   \dimen@\eql@marginleft@
2116   \advance\dimen@\eql@tagwidth@
2117   \eql@adjust@try\dimen@
2118   \ifnum\badness<\eql@marginbadness@
2119     \eql@adjust@print@alignright\eql@marginleft@\eql@tagwidth@
2120     \eql@tagbox@print@right
2121   \else
2122     \ifdim\eql@flushleftmarginmin<\eql@marginleft@
2123       \dimen@\eql@flushleftmarginmin@
2124       \advance\dimen@\eql@tagwidth@
2125       \eql@adjust@try\dimen@
2126     \fi
2127     \ifnum\badness<\eql@maxbadness@
2128       \eql@adjust@print@alignright\eql@flushleftmarginmin@\eql@tagwidth@
2129       \eql@tagbox@print@right
2130     \else
2131       \eql@adjust@flushleft@shoveright@notag
2132       \eql@tagbox@print@right@raise
2133     \fi
2134   \fi
2135 }
```

TODO: describe

```
2136 \def\eql@adjust@flushleft@tagsleft@shoveright@tag{%
2137   \ifdim\eql@tagwidth<\eql@flushleftmarginmin@
2138     \eql@tagbox@print@left
2139     \eql@adjust@flushleft@shoveright@notag
2140   \else
2141     \ifdim\eql@tagwidth<\eql@marginleft@
2142       \eql@adjust@try\eql@marginleft@
2143       \ifnum\badness<\eql@marginbadness@
2144         \eql@tagbox@print@left
2145         \eql@adjust@print@alignright\eql@marginleft@\z@
2146       \else
```

```

2147     \eql@adjust@try\eql@tagwidth@
2148     \ifnum\badness<\eql@maxbadness@
2149         \eql@tagbox@print@left
2150         \eql@adjust@print@alignright\eql@tagwidth@\z@
2151     \else
2152         \eql@tagbox@print@left@raise
2153         \eql@adjust@print@alignright\eql@flushleftmarginmin@\z@
2154     \fi
2155 \fi
2156 \else
2157     \ifdim\eql@tagwidth@>\eql@flushleftmarginmax@
2158         \eql@tagbox@print@left@raise
2159         \eql@adjust@flushleft@shoveright@notag
2160     \else
2161         \eql@adjust@try\eql@tagwidth@
2162         \ifnum\badness<\eql@maxbadness@
2163             \eql@tagbox@print@left
2164             \eql@adjust@print@alignright\eql@tagwidth@\z@
2165         \else
2166             \eql@tagbox@print@left@raise
2167             \eql@adjust@flushleft@shoveright@notag
2168         \fi
2169     \fi
2170 \fi
2171 \fi
2172 \eql@display@firstavail@set\z@
2173 }

2174 \def\eql@adjust@flushleft@shovecenter{%
2175     \eql@error{shove center not implemented for left alignment}%
2176 }
2177 \let\eql@adjust@flushleft@tagsright@shovecenter@notag
2178     \eql@adjust@flushleft@shovecenter
2179 \let\eql@adjust@flushleft@tagsright@shovecenter@tag
2180     \eql@adjust@flushleft@shovecenter
2181 \let\eql@adjust@flushleft@tagsleft@shovecenter@notag
2182     \eql@adjust@flushleft@shovecenter
2183 \let\eql@adjust@flushleft@tagsleft@shovecenter@tag
2184     \eql@adjust@flushleft@shovecenter

```

L Single-Line Equation

TODO: describe

L.1 Environment

```

2185 \def\eql@single@cr{%
2186     \eql@error{Cannot use ‘\string\’ within display equation.
2187     Please switch to equations environment}%
2188 }

```

TODO: describe

```

2189 \def\eql@single@start{%
2190     \eql@halign@catchprevdepth
2191     \eql@tagging@start
2192     $$$$$$

```

```

2193 \eql@numbering@eval@mode
2194 \let\eql@numbering@subeq@use\eql@false
2195 \eql@stack@save@single

```

TODO: make other display environments push these!?

```

2196 \eql@numbering@single@init
2197 \ifdefined\eql@single@native
2198   \let\eql@single@start@sel\eql@single@start@native
2199   \let\eql@single@end@sel\eql@single@end@native
2200   \let\raisetag\eql@raisetag@default
2201 \else
2202   \let\eql@single@start@sel\eql@single@start@adjust
2203   \let\eql@single@end@sel\eql@single@end@adjust
2204 \fi
2205 \ifdefined\eql@single@crerror\else
2206   \let\\eql@single@cr
2207 \fi
2208 \eql@single@start@sel
2209 }

2210 \def\eql@single@end{%
2211   \eql@punct@apply@block
2212   \eql@hook@eqout
2213   \eql@single@end@sel
2214   \eql@stack@restore
2215   $$$$$$
2216   \eql@tagging@end
2217   \eql@halign@leave
2218 }

```

TODO: : try to feed in tagging after catchprevdepth

```

2219 \def\eql@single@main{%
2220   \expandafter\eql@single@start
2221   \eql@scan@body
2222   \eql@single@end
2223 }

```

L.2 Native

```

2224 \def\eql@single@start@native{%
2225 % \mathopen{}%
2226 \eql@hook@eqin
2227 }%

```

TODO: describe

```

2228 \def\eql@single@end@native{%
2229 % \mathclose{}%
2230 \if@eqnsw
2231   \ifdefined\eql@tagsleft
2232     \leqno
2233   \else
2234     \eqno
2235   \fi
2236   \eql@compose@print
2237 \fi
2238 \ifnum\eql@displaybreak@open@=\@MM\else
2239   \postdisplaypenalty\eql@displaybreak@open@
2240 \fi
2241 }%

```

L.3 Adjustment

```
2242 \def\eql@single@start@adjust{%
2243   \eql@totalrows@\@ne
2244   \eql@row@\z@
2245   \eql@display@init
2246   \let\shoveleft\eql@adjust@shoveleft
2247   \let\shovecenter\eql@adjust@shovecenter
2248   \let\shoveright\eql@adjust@shoveright
2249   \ifdefined\eql@flushleft\else
2250     \eql@marginleft@\z@
2251     \eql@marginright@\z@
2252   \fi
2253   \eql@adjust@init
2254   \eql@shape@eval

2255   \eql@row@\@ne
2256   \setbox\eql@fieldbox@\hbox\bgroup
2257     \eql@restore@hfuzz
2258     \eql@strut@field
2259     $\m@th\displaystyle%$
2260     \eql@hook@eqin
2261 }

2262 \def\eql@single@end@adjust{%
2263   \eql@tagging@mathsave
2264   $%$
2265   \hfil
2266   \kern\z@
2267   \egroup

2268   \eql@fieldwidth@\wd\eql@fieldbox@
2269   \eql@line@height@\ht\eql@fieldbox@
2270   \eql@line@depth@\dp\eql@fieldbox@

2271   \eql@halign@init{}}%
2272   \halign{##\cr
2273     \noalign{\eql@halign@before}}%
2274     \if@eqnsw
2275       \eql@tagbox@make\eql@compose@print
2276       \eql@adjust@sel@tag
2277     \else
2278       \eql@adjust@sel@notag
2279     \fi
2280     \cr
2281     \noalign{\eql@halign@after}}%
2282   \eql@tagging@tablesavelines
2283 }%
2284 \eql@row@\tw@
2285 \eql@display@close
2286 }
```

M Multi-Line Support

TODO: describe

M.1 Registers

```
\eql@column@
\eql@totalcolumns@
```

```

2287 \newcount\eql@column@
2288 \newcount\eql@totalcolumns@

```

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

```

2289 \newdimen\eql@totalwidth@

```

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

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

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

```

2290 \newdimen\eql@linewidth@
2291 \newdimen\eql@line@avail@
2292 \newdimen\eql@line@pos@

```

`\eql@fieldlength@tab`

`\eql@fieldlength@save`

`\eql@fieldlength@get`

```

2293 \let\eql@fieldlength@tab@empty
2294 \def\eql@fieldlength@save#1{%
2295   \begingroup
2296     \let\or\relax
2297     \global\edef\eql@fieldlength@tab{%
2298       \eql@fieldlength@tab
2299       \ifnum#1=\@ne
2300         \or
2301         \else
2302           ,%
2303         \fi
2304       \the\wd\eql@fieldbox@
2305     }%
2306   \endgroup
2307 }
2308 \def\eql@fieldlength@get#1{%
2309   \ifcase\expandafter#1\eql@fieldlength@tab\fi
2310 }

```

`\eql@tagwidth@get`

`\eql@tagwidth@save`

```

2311 \let\eql@tagwidth@tab@empty
2312 \def\eql@tagwidth@get#1{%
2313   \ifcase\expandafter#1\eql@tagwidth@tab\fi
2314 }
2315 \def\eql@tagwidth@save{%
2316   \begingroup
2317     \let\or\relax
2318     \global\edef\eql@tagwidth@tab{\eql@tagwidth@tab\or\the\eql@tagwidth@}%
2319   \endgroup
2320 }
2321 \def\eql@tagwidth@savezero{%
2322   \begingroup
2323     \let\or\relax
2324     \global\edef\eql@tagwidth@tab{\the\eql@tagwidth@\eql@tagwidth@tab}%
2325   \endgroup
2326 }

```

M.2 Measure Support

TODO: describe

```

2327 \def\eql@measure@init#1{%

```

```

2328 \measuring@true
2329 \eql@row@z@
2330 \let\displaybreak\eql@displaybreak@measure
2331 \tabskip\z@skip
2332 \everycr{%
2333   \noalign{%
2334     \global\advance\eql@row@\@ne
2335     #1%
2336   }%
2337 }%
2338 }

```

sure@restorecounters

measure@savecounters

```

2339 \let\eql@measure@restorecounters\@empty
2340 \def\eql@measure@savecounters{%
2341   \begingroup
2342     \def\@elt##1{%
2343       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2344     \global\edef\@gtempa{%
2345       \cl@ckpt
2346       \let\noexpand\eql@measure@restorecounters\noexpand\@empty
2347     }%
2348   \endgroup
2349   \let\eql@measure@restorecounters\@gtempa
2350 }

```

M.3 Print Support

TODO: describe

eql@print@inithalign

```

2351 \def\eql@print@init#1{%
2352   \eql@row@z@
2353   \eql@halign@init{%
2354     \global\eql@displaybreak@pen@\@MM
2355     \global\advance\eql@row@\@ne
2356     #1%
2357   }%
2358 }

2359 \def\eql@print@overfull{%
2360   \dimen@eql@line@width@
2361   \advance\dimen@-\hfuzz
2362   \ifdim\dimen@>\displaywidth
2363     \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
2364     \wd\z@\z@
2365     \ht\z@\eql@line@height@
2366     \dp\z@\eql@line@depth@
2367     \box\z@
2368   \fi
2369 }

```

l@tagbox@print@multi

```

2370 \def\eql@tagbox@print@multi{%
2371   \advance\eql@tagwidth@-\eql@tagfuzz@

```

```

2372 \ifdefined\eq\@tagsleft
2373   \eq\@display\@firstavail\@set\z\@
2374   \ifdim\eq\@tagwidth\@>\eq\@line\@avail\@
2375     \eq\@tagbox\@print\@left\@raise
2376   \else
2377     \eq\@tagbox\@print\@left
2378   \fi
2379   \kern\displaywidth
2380 \else
2381   \kern\displaywidth
2382   \advance\eq\@tagwidth\@ \eq\@line\@width\@
2383   \ifdim\eq\@tagwidth\@>\displaywidth
2384     \eq\@tagbox\@print\@right\@raise
2385   \else
2386     \eq\@tagbox\@print\@right
2387   \fi
2388 \fi
2389 }

```

M.4 Line Breaks

TODO: describe

`\eq\@math\@cr`

```

2390 \protected\def\eq\@math\@cr{%
2391   \eq\@amp\protecttwo\eq\@test\staropt\@tight
2392   {\global\eq\@displaybreak\@open\@M\eq\@math\@cr\@}\eq\@math\@cr\@z\@}

```

`\eq\@math\@cr\@`

```

2393 \def\eq\@math\@cr\@[#1]{%
2394   \eq\@math\@cr\@@@
2395   \cr
2396   \noalign{%
2397     \ifnum\eq\@displaybreak\@pen\@=\@MM
2398       \penalty\interdisplaylinepenalty
2399     \else
2400       \penalty\eq\@displaybreak\@pen\@
2401     \fi
2402     \advance\eq\@vspaceskip\@ \glueexpr#1\relax%
2403     \vskip\eq\@vspaceskip\@
2404     \global\eq\@vspaceskip\@z\@skip
2405   }%
2406 }

```

`\eq\@let\@cr`

```

2407 \def\eq\@let\@cr#1{%
2408   \let\\\eq\@math\@cr
2409   \let\eq\@math\@cr\@@@#1%
2410 }

```

M.5 Intertext

TODO: describe

TODO: revert in everymath?

```
2411 \def\eq@intertext@default{\eq@error{Invalid use of \string\intertext}}
2412 \eq@amsmath@let\intertext\eq@intertext@default
```

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

```
2413 \def\eq@intertext@process{%
2414   \eq@math@cr@@@
2415   \cr
2416   \ifmeasuring@
2417     \expandafter\@gobble
2418   \else
2419     \expandafter\eq@intertext@print
2420   \fi
2421 }
```

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

```
2422 \def\eq@intertext@print#1{%
2423   \noalign{%
2424     \eq@halign@after
2425     \let\eq@skip@force@below\z@
2426     \let\eq@skip@force@above\z@
2427     \eq@setkeys{intertext}\eq@intertext@opt
2428     \openup-\eq@spread@amount@
2429     \penalty\postdisplaypenalty
2430     \ifcase\eq@skip@force@below\relax
2431       \advance\eq@vspaceskip@\glueexpr\eq@skip@long@below\relax
2432     \or
2433       \advance\eq@vspaceskip@\glueexpr\eq@skip@short@below\relax
2434     \or
2435       \advance\eq@vspaceskip@\glueexpr\eq@skip@cont@below\relax
2436     \or
2437       \advance\eq@vspaceskip@\glueexpr\eq@skip@par@below\relax
2438     \or
2439       \advance\eq@vspaceskip@\glueexpr\eq@skip@top@below\relax
2440     \or
2441       \advance\eq@vspaceskip@\z@skip
2442     \or
2443       \advance\eq@vspaceskip@\glueexpr\eq@skip@med@below\relax
2444     \or
2445       \advance\eq@vspaceskip@\glueexpr\eq@skip@custom@below\relax
2446     \fi
2447     \vskip\eq@vspaceskip@
2448     \global\eq@vspaceskip@\z@skip
2449     \vbox{%
2450       \@parboxrestore
2451       \ifdim
2452         \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2453       \else
2454         \parshape\@ne
2455         \@totalleftmargin\linewidth
2456       \fi
2457       \noindent
2458       \ignorespaces
2459       #1%
2460     \par
2461   }%

```

```

2462 \penalty\predisplaypenalty
2463 \ifcase\eq@skip@force@above\relax
2464 \vskip\glueexpr\eq@skip@long@above\relax
2465 \or
2466 \vskip\glueexpr\eq@skip@short@above\relax
2467 \or
2468 \vskip\glueexpr\eq@skip@cont@above\relax
2469 \or
2470 \vskip\glueexpr\eq@skip@par@above\relax
2471 \or
2472 \vskip\glueexpr\eq@skip@top@above\relax
2473 \or
2474 \vskip\z@skip
2475 \or
2476 \vskip\glueexpr\eq@skip@med@above\relax
2477 \or
2478 \vskip\glueexpr\eq@skip@custom@above\relax
2479 \fi
2480 % \eq@halign@prevdepth@\maxdimen
2481 \eq@halign@prevdepth@\z@
2482 \eq@halign@before
2483 }
2484 }

```

TODO: describe

```

2485 \newenvironment{eq@intertext}{%
2486 \eq@testopt@tight\eq@intertext@{}%
2487 }{%
2488 \aftergroup\eq@intertext@after
2489 \ignorespacesafterend
2490 }

```

TODO: describe

```

2491 \def\eq@intertext@env{intertext}
2492 \def\eq@intertext@[#1]{%
2493 \global\def\eq@intertext@opt{#1}%
2494 \ifx\@currentvir\eq@intertext@env
2495 \expandafter\eq@scan@env\expandafter\eq@intertext@inject
2496 \else
2497 \expandafter\eq@intertext@process
2498 \fi
2499 }

```

TODO: describe

```

2500 \def\eq@intertext@inject{%
2501 \global\edef\eq@intertext@after{%
2502 \noexpand\eq@intertext@process{%
2503 \ifx\eq@scan@body\eq@scan@body@dump
2504 \eq@scan@body@dump
2505 \else
2506 \noexpand\scantokens{\eq@scan@body@dump}%
2507 \fi
2508 }%
2509 }%
2510 }

```

M.6 Main

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

`\eql@multi@main`

```
2511 \let\eql@multi@mode@lines\eql@false
2512 \def\eql@multi@main{%
2513   \eql@halign@catchprevdepth
2514   \eql@tagging@start
2515   $$$$$
2516   \eql@numbering@eval@mode
2517   \eql@stack@save@multi
2518   \ifdefined\eql@subequations@active
2519     \let\eql@numbering@subeq@use\eql@false
2520   \fi
2521   \ifdefined\eql@numbering@subeq@use
2522     \eql@numbering@subeq@init
2523   \fi
2524   \let\intertext\eql@intertext
2525   \let\endintertext\endeql@intertext
2526   \let\shoveleft\eql@adjust@shoveleft
2527   \let\shovecenter\eql@adjust@shovecenter
2528   \let\shoveright\eql@adjust@shoveright
2529   \ifdefined\eql@multi@mode@lines
2530     \expandafter\eql@lines@measure
2531   \else
2532     \ifdefined\eql@ampproof@active
2533       \eql@ampproof
2534     \fi
2535     \expandafter\eql@align@measure
2536   \fi
2537   \ifx\eql@numbering@subeq@use\@ne
2538     \eql@numbering@subeq@revert
2539   \fi
2540   \ifdefined\eql@multi@mode@lines\else
2541     \ifdefined\eql@multi@lines@fallback
2542       \ifnum\eql@totalcolumns@=\@ne
2543         \let\eql@multi@mode@lines\eql@true
2544         \eql@shape@set{r}%
2545       \eql@lines@measure
2546     \fi
2547   \fi
2548   \fi
2549   \ifdefined\eql@multi@mode@lines
2550     \expandafter\eql@lines@print
2551   \else
2552     \expandafter\eql@align@print
2553   \fi
2554   \ifdefined\eql@numbering@subeq@use
2555     \eql@numbering@subeq@close
2556   \fi
2557   \eql@stack@restore
2558   $$$$$
2559   \eql@tagging@end
2560   \eql@halign@leave
2561 }
```

TODO: describe

```
2562 \def\eql@mode@equation{%
2563   \let\eql@equations@mode@single\eql@true
2564   \ifdefined\eql@single@doscan
2565     \let\eql@equations@main\eql@single@main
2566     \let\eql@equations@end\@empty
2567   \else
2568     \let\eql@equations@main\@undefined
2569     \let\eql@equations@end\eql@single@end
2570   \fi
2571 }
2572 \def\eql@mode@align{%
2573   \let\eql@equations@mode@single\eql@false
2574   \let\eql@multi@mode@lines\eql@false
2575   \let\eql@equations@main\eql@multi@main
2576   \let\eql@equations@end\@empty
2577 }
2578 \def\eql@mode@lines{%
2579   \let\eql@equations@mode@single\eql@false
2580   \let\eql@multi@mode@lines\eql@true
2581   \let\eql@equations@main\eql@multi@main
2582   \let\eql@equations@end\@empty
2583 }
2584 \eql@mode@align
```

N Multi-Line Lines Mode

N.1 Measure

TODO: describe

```
2585 \def\eql@lines@measure@line@begin{%
2586 (dev)\eql@dev{starting line \the\eql@row}%
2587   \eql@numbering@measure@line@begin
2588   \eql@hook@linein
2589 }
```

TODO: describe

```
2590 \def\eql@lines@measure@line@end{%
2591   \eql@punct@apply@line
2592   \eql@hook@lineout
2593 }
```

TODO: describe

```
2594 \def\eql@lines@measure@field{%
2595   \kern\wd\eql@fieldbox@
2596 }
```

TODO: describe

```
2597 \def\eql@lines@measure@tag{%
2598   \ifnum\eql@numbering@target@<\z@
2599     \if@eqnsw
2600       \eql@compose@null
2601     \fi
2602   \fi
```

2603 }

`\eql@lines@measure`

```
2604 \def\eql@lines@measure{%
2605 (dev)\eql@dev@enter\eql@lines@measure
2606 \eql@measure@savecounters
2607 \setbox\z@\vbox{%
2608 \eql@numbering@measure@init
2609 \eql@measure@init\eql@lines@measure@line@begin
2610 \eql@let@cr\eql@lines@measure@line@end
2611 \halign{%
2612 \setbox\eql@fieldbox@\hbox{%
2613 \@lign
2614 $\m@th\displaystyle
2615 \eql@hook@colin
2616 ##%
2617 \eql@punct@apply@col
2618 \eql@hook@colout
2619 $%
2620 }%
2621 \eql@lines@measure@field
2622 \eql@lines@measure@tag
2623 \crr
2624 \noalign{%
2625 \eql@hook@blockbefore
2626 }%
2627 \eql@hook@blockin
2628 \eql@scan@body
2629 \ifvmode\else
2630 \eql@punct@apply@block
2631 \eql@hook@blockout
2632 \eql@lines@measure@line@end
2633 \cr
2634 \fi
2635 \omit
2636 \cr
2637 \noalign{%
2638 \eql@hook@blockafter
2639 }%
2640 }%
2641 \global\advance\eql@row@-\tw@
2642 \eql@numbering@measure@eval
2643 }%
2644 \eql@totalrows@\eql@row@
2645 \setbox\z@\vbox{%
2646 \unvbox\z@
2647 \unpenalty
2648 \global\setbox\@ne\lastbox
2649 }%
2650 \eql@totalwidth@\wd\@ne
2651 \ifdefined\eql@numbering@subeq@use
2652 \eql@numbering@subeq@test
2653 \fi
2654 \eql@measure@restorecounters
2655 (dev)\eql@dev@leave\eql@lines@measure
2656 }
```

N.2 Print

TODO: describe

`\eql@lines@math@cr`

```
2657 \def\eql@lines@print@line@begin{%
2658 (dev)\eql@dev{starting line \the\eql@row}%
2659 \eql@numbering@print@line@begin
2660 \eql@hook@linein
2661 }
```

TODO: describe

```
2662 \def\eql@lines@print@line@end{%
2663 \eql@punct@apply@line
2664 \eql@hook@lineout
2665 }
```

TODO: describe

```
2666 \def\eql@lines@print@line@adjust{%
2667 \eql@numbering@print@line@eval
2668 \eql@fieldwidth@\wd\eql@fieldbox@
2669 \eql@line@height@\ht\eql@fieldbox@
2670 \eql@line@depth@\dp\eql@fieldbox@
2671 \if@eqnsw
2672 \eql@tagbox@make\eql@compose@print
2673 \eql@adjust@sel@tag
2674 \else
2675 \eql@adjust@sel@notag
2676 \fi
2677 }
```

TODO: describe

```
2678 \def\eql@lines@print{%
2679 (dev)\eql@dev@enter\eql@lines@print
2680 \eql@display@init
2681 \ifdefined\eql@flushleft
2682 % \TODO any init needed here? marginleft is used per line!
2683 \else
2684 \ifdefined\eql@paddingmax
2685 \eql@marginleft@\z@
2686 \eql@marginright@\z@
2687 \else
2688 \dimen@\displaywidth
2689 \advance\dimen@-\eql@totalwidth@
2690 \advance\dimen@-\eql@tagmargin@
2691 \divide\dimen@\tw@
2692 \eql@marginleft@\dimen@
2693 \advance\eql@marginleft@-\glueexpr\eql@paddingleft@val\relax
2694 \ifdim\eql@marginleft@<\z@
2695 \eql@marginleft@\z@
2696 \fi
2697 \eql@marginright@\dimen@
2698 \advance\eql@marginright@-\glueexpr\eql@paddingright@val\relax
2699 \ifdim\eql@marginright@<\z@
2700 \eql@marginright@\z@
2701 \fi
```

```

2702   \fi
2703   \fi
2704   \eql@adjust@init
2705   \eql@numbering@print@init
2706   \eql@print@init\eql@lines@print@line@begin
2707   \eql@let@cr\eql@lines@print@line@end
2708   \tabskip\z@skip
2709   \halign{%
2710     \eql@shape@eval
2711     \setbox\eql@fieldbox@\hbox{%
2712       \eql@restore@hfuzz
2713       \eql@strut@field
2714       \@lign
2715       $\m@th\displaystyle
2716       \eql@hook@colin
2717       ##%
2718       \eql@punct@apply@col
2719       \eql@hook@colout
2720       \eql@tagging@mathsave
2721       $%
2722       \hfil
2723       \kern\z@
2724     }%
2725     \eql@lines@print@line@adjust
2726     \crr
2727     \noalign{%
2728       \eql@halign@before
2729       \eql@numbering@print@block@begin
2730       \eql@hook@blockbefore
2731     }%
2732 % \TODO relax? leavevmode?!
2733   \eql@hook@blockin
2734   \eql@scan@body
2735   \ifvmode\else
2736     \eql@punct@apply@block
2737     \eql@hook@blockout
2738     \eql@lines@print@line@end
2739     \cr
2740   \fi
2741   \noalign{%
2742     \eql@hook@blockafter
2743     \eql@halign@after
2744 (dev)\eql@dev@leave\eql@lines@print
2745   }%
2746   \eql@tagging@tablesavelines
2747 }%
2748 \eql@display@close
2749 }

```

O Multi-Line Align Mode

TODO: describe

O.1 Registers

TODO: describe

```
2750 \let\eql@align@margin@eql@true
```

```
\eql@align@inter@
```

```
2751 \newcount\eql@align@inter@
```

```
\eql@colwidth@tab
```

```
2752 \let\eql@colwidth@tab@empty
```

```
\eql@align@colwidth@get
```

```
2753 \def\eql@align@colwidth@get#1{%
2754   \ifcase\expandafter#1\eql@colwidth@tab\else\z@fi
2755 }
2756 \def\eql@align@colwidth@save{%
2757   \begingroup
2758     \let\or\relax
2759     \global\edef\eql@colwidth@tab{\or\the\wd\thr@}\eql@colwidth@tab}%
2760   \endgroup
2761 }
```

O.2 General Processing, Preamble

TODO: describe

```
\eql@align@add@amp
```

```
\eql@align@completerow
```

```
2762 \def\eql@align@add@amp#1{\if m#1&\omit\expandafter\eql@align@add@amp\fi}
2763 \def\eql@align@completerow#1{%
2764   \begingroup
2765   \count@#1%
2766   \advance\count@-\eql@column@
2767   \advance\count@\@ne
2768   \edef\eql@tmp{\endgroup
2769     \expandafter\eql@align@add@amp\romannumeral\number\count@ 000q}%
2770   \eql@tmp
2771 }
```

```
2772 \def\eql@align@print@trailright{%
2773   &\omit
2774   \global\advance\eql@column@\@ne
2775   \setbox\eql@fieldbox@\hbox{%
2776     \kern-\wd\eql@fieldbox@\box\eql@fieldbox@
2777   }%
2778   \eql@align@print@field
2779 }
```

O.3 Print

TODO: describe

```
2780 \def\eql@align@print@field{%
```

determine available and used space

```
2781 \dimen@eql@align@colwidth@get\eql@column@\relax
2782 \ifdim\wd\eql@fieldbox@>\z@
2783 \ifdim\eql@line@width@=\z@
2784 \eql@line@avail@\eql@line@pos@
2785 \ifodd\eql@column@
2786 \advance\eql@line@avail@\dimen@
2787 \advance\eql@line@avail@-\wd\eql@fieldbox@
2788 \fi
2789 \global\eql@line@avail@\eql@line@avail@
2790 \fi
2791 \eql@line@width@\eql@line@pos@
2792 \ifodd\eql@column@
2793 \advance\eql@line@width@\dimen@
2794 \else
2795 \advance\eql@line@width@\wd\eql@fieldbox@
2796 \fi
2797 \global\eql@line@width@\eql@line@width@
2798 \fi
2799 \advance\eql@line@pos@\dimen@
2800 \ifodd\eql@column@\else
2801 \advance\eql@line@pos@\eql@colsep@
2802 \fi
2803 \global\eql@line@pos@\eql@line@pos@
```

update height and depth

```
2804 \ifdim\ht\eql@fieldbox@>\eql@line@height@
2805 \global\eql@line@height@\ht\eql@fieldbox@
2806 \fi
2807 \ifdim\dp\eql@fieldbox@>\eql@line@depth@
2808 \global\eql@line@depth@\dp\eql@fieldbox@
2809 \fi
```

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

```
2810 %
2811 % \box\eql@fieldbox@
2812 %
2813 % \dimen@\eql@align@colwidth@get\eql@column@\relax
2814 % \advance\dimen@-\wd\eql@fieldbox@
2815 % \ifodd\eql@column@
2816 % \kern\dimen@
2817 % \box\eql@fieldbox@
2818 % \else
2819 % \box\eql@fieldbox@
2820 % \kern\dimen@
2821 % \fi
2822 %
2823 \dimen@\eql@align@colwidth@get\eql@column@\relax
2824 \ifodd\eql@column@
2825 \kern\dimen@
2826 \else
2827 \advance\dimen@-\wd\eql@fieldbox@
2828 \box\eql@fieldbox@
2829 \kern\dimen@
2830 \fi
2831 %
```

2832 }

align@print@line@begin

```
2833 \def\eq@align@print@line@begin{%
2834 (dev)\eq@dev{starting line \the\eq@row}%
2835 \global\eq@column@z@
2836 \global\eq@line@pos@\eq@marginleft@
2837 \global\eq@line@width@z@
2838 \global\eq@line@avail@\eq@totalwidth@
2839 \global\eq@line@height@z@
2840 \global\eq@line@depth@z@
2841 \eq@numbering@print@line@begin
2842 \eq@hook@linein
2843 }
```

align@print@line@end

```
2844 \def\eq@align@print@line@end{%
2845 \eq@punct@apply@line
2846 \eq@hook@lineout
2847 % \TODO add an even column with empty stuff if box processing deferred
2848 \ifodd\eq@column@
2849 \expandafter\eq@align@print@trailright
2850 \fi
2851 \eq@align@completerow\eq@totalcolumns@
2852 \eq@align@print@line
2853 }
```

eq@align@print@line

```
2854 \def\eq@align@print@line{%
2855 \dimen@\eq@totalwidth@
2856 \advance\dimen@\eq@colsep@
2857 \kern-\dimen@

determine first line available space

2858 \eq@display@firstavail@set\eq@line@avail@
2859 \eq@print@overfull
2860 \eq@numbering@print@line@eval
2861 \if@eqnsw
2862 \eq@tagbox@make\eq@compose@print
2863 \eq@tagging@tagaddbox
2864 \eq@tagbox@print@multi
2865 \else
2866 \eq@tagging@tagaddbox
2867 \kern\displaywidth
2868 \fi
2869 }
```

\eq@align@print

```
2870 \def\eq@align@print{%
2871 (dev)\eq@dev@center\eq@align@print
2872 \eq@align@adjust
2873 \eq@display@init
2874 \eq@numbering@print@init
2875 \eq@print@init\eq@align@print@line@begin
2876 \eq@let@cr\eq@align@print@line@end
```

```

2877 \tabskip\eq@marginleft@
2878 \halign{%
2879   &%
2880   \global\advance\eq@column@\@ne
2881   \hfil
2882   \global\setbox\eq@fieldbox@\hbox{%
2883     \eq@strut@field
2884     \@lign
2885     $\m@th\displaystyle
2886     \eq@hook@colin
2887     ##%
2888     \eq@class@innerleft
2889     \eq@hook@innerleft
2890     \eq@tagging@mathsave
2891     $%
2892     \eq@tagging@mathaddlast
2893   }%
2894   \global\eq@fieldwidth@\wd\eq@fieldbox@
2895   \eq@align@print@field
2896   \tabskip\z@skip
2897   &%
2898   \global\advance\eq@column@\@ne
2899   \setbox\eq@fieldbox@\hbox{%
2900 % \TODO printing left field in right field
2901   \kern-\wd\eq@fieldbox@
2902   \box\eq@fieldbox@
2903   \eq@strut@field
2904   \@lign
2905   $\m@th\displaystyle
2906   \eq@hook@innerright
2907   \eq@class@innerright@sel
2908   ##%
2909   \eq@punct@apply@col
2910   \eq@hook@colout
2911   \eq@tagging@mathsave
2912   $%
2913   \eq@tagging@mathaddlast
2914   }%
2915   \eq@align@print@field
2916   \hfil
2917   \tabskip\eq@colsep@\relax
2918 \crrc
2919 \noalign{%
2920   \eq@halign@before
2921   \eq@numbering@print@block@begin
2922   \eq@hook@blockbefore
2923 }%
2924 \eq@hook@blockin
2925 \eq@scan@body
2926 \ifvmode\else
2927   \eq@punct@apply@block
2928   \eq@hook@blockout
2929   \eq@align@print@line@end
2930 \cr
2931 \fi
2932 \noalign{%
2933   \eq@hook@blockafter
2934   \eq@halign@after

```

```

2935 (dev)\eql@dev@leave\eql@align@print
2936   }%
2937   \eql@tagging@tables@save@align
2938 }%
2939 \eql@display@close
2940 }

```

O.4 Measure

TODO: describe

n@measure@line@begin

```

2941 \def\eql@align@measure@line@begin{%
2942 (dev)\eql@dev{starting line \the\eql@row}%
2943 \global\eql@column@z@
2944 \eql@numbering@measure@line@begin
2945 \eql@hook@linein
2946 }

2947 \def\eql@align@measure@field{%
2948 \eql@fieldlength@save\eql@column@
2949 \kern\wd\eql@fieldbox@
2950 }

```

ign@measure@line@end

```

2951 \def\eql@align@measure@line@end{%
2952 \eql@punct@apply@line
2953 \eql@hook@lineout
2954 &\omit
2955 \ifnum\eql@column@>\eql@totalcolumns@
2956 \global\eql@totalcolumns@\eql@column@
2957 \fi
2958 \eql@align@measure@tag
2959 }

```

ql@align@measure@tag

```

2960 \def\eql@align@measure@tag{%
2961 \ifnum\eql@numbering@target@<z@
2962 \if@eqnsw
2963 \eql@tagbox@make\eql@compose@measure
2964 \else
2965 \eql@tagwidth@z@
2966 \fi
2967 \eql@tagwidth@save
2968 \fi
2969 }

```

\eql@align@measure

```

2970 \def\eql@align@measure{%
2971 (dev)\eql@dev@center\eql@align@measure
2972 \eql@totalcolumns@z@
2973 \let\eql@tagwidth@tab@empty
2974 \let\eql@fieldlength@tab@empty
2975 \eql@measure@savecounters

```

```

2976 \setbox\z@\vbox{%
2977   \eq\@numbering@measure@init
2978   \eq\@measure@init\eq\@align@measure@line@begin
2979   \eq\@let@cr\eq\@align@measure@line@end
2980   \tabskip\z@skip
2981   \halignf%
2982     &%
2983     \global\advance\eq\@column@\@ne
2984     \hfil
2985     \global\setbox\eq\@fieldbox@\hbox{%
2986       \@lign
2987       $\m@th\displaystyle
2988       \eq\@hook@colin
2989       ##%
2990       \eq\@class@innerleft
2991       \eq\@hook@innerleft
2992       $%
2993     }%
2994     \global\eq\@fieldwidth@\wd\eq\@fieldbox@
2995     \eq\@align@measure@field
2996     &%
2997     \global\advance\eq\@column@\@ne
2998     \setbox\eq\@fieldbox@\hbox{%
2999       \@lign
3000       $\m@th\displaystyle
3001       \eq\@hook@innerright
3002       \eq\@class@innerright@sel
3003       ##%
3004       \eq\@punct@apply@col
3005       \eq\@hook@colout
3006       $%
3007     }%
3008     \eq\@align@measure@field
3009     \hfil
3010   \crcr
3011   \noalignf%
3012     \eq\@hook@blockbefore
3013   }%
3014   \eq\@hook@blockin
3015   \eq\@scan@body

```

TODO: test for vmode okay?!

```

3016   \ifvmode\else
3017     \eq\@punct@apply@block
3018     \eq\@hook@blockout
3019     \eq\@align@measure@line@end
3020     \cr
3021   \fi
3022   \noalignf%
3023     \eq\@hook@blockafter
3024   }%

```

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

```

3025   \eq\@align@completerow\eq\@totalcolumns@
3026   \cr
3027 }%
3028 \global\advance\eq\@row@-\tw@

```

```

3029 \eql@numbering@measure@eval
3030 \ifnum\eql@numbering@target@>\z@
3031 \eql@tagbox@make\eql@compose@measure
3032 \eql@tagwidth@savezero
3033 \fi
3034 }%
3035 \eql@totalrows@\eql@row@
3036 \ifdefined\eql@numbering@subeq@use
3037 \eql@numbering@subeq@test
3038 \fi
3039 \eql@measure@restorecounters
3040 % \eql@totalwidth@\wd\z@

3041 \setbox\z@\vbox{%
3042 \unvbox\z@
3043 \unpenalty
3044 \global\setbox\@ne\lastbox
3045 }%

3046 % \TODO or this one?!
3047 \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

3048 \let\eql@colwidth@tab\@empty
3049 \loop
3050 \setbox\@ne\hbox{%
3051 \unhbox\@ne
3052 \unskip
3053 \global\setbox\thr@\@lastbox
3054 }%
3055 \ifhbox\thr@@
3056 \eql@align@colwidth@save
3057 \repeat
3058 (dev)\eql@dev@leave\eql@align@measure
3059 }

```

O.5 Adjust

TODO: describe

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

```

3060 \def\eql@align@adjust{%
3061 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
3062 \eql@colsepmax@\glueexpr\eql@colsepmax@val\relax

```

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

```

3063 \ifodd\eql@totalcolumns@
3064 \advance\eql@totalcolumns@\@ne
3065 \fi

```

TODO: should we guard against no columns?!

```

3066 \ifnum\eql@totalcolumns@<\thr@@
3067 \let\eql@align@margins\eql@true
3068 \fi

```

```

3069 \eql@align@inter@\eql@totalcolumns@
3070 \divide\eql@align@inter@\tw@
3071 \advance\eql@align@inter@\m@ne

3072 \eql@colsep@\displaywidth
3073 \advance\eql@colsep@-\eql@totalwidth@
3074 \ifdefined\eql@flushleft
3075 \advance\eql@colsep@-\eql@flushleftmargin@
3076 \fi
3077 \count@\eql@align@inter@
3078 \ifdefined\eql@align@argins
3079 \ifdefined\eql@flushleft
3080 \advance\count@\@ne
3081 \else
3082 \advance\count@\tw@
3083 \fi
3084 \fi
3085 \divide\eql@colsep@\count@

```

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

```

3086 \ifdim\eql@colsep@<\eql@colsepmin@
3087 \eql@colsep@\eql@colsepmin@
3088 \else
3089 \ifdim\eql@colsep@>\eql@colsepmax@
3090 \eql@colsep@\eql@colsepmax@
3091 \fi
3092 \fi

3093 \ifdefined\eql@flushleft
3094 \ifdim\eql@colsep@=\eql@colsepmin@
3095 \eql@marginleft@\displaywidth
3096 \advance\eql@marginleft@-\eql@totalwidth@
3097 \advance\eql@marginleft@-\eql@align@inter@\eql@colsep@
3098 \ifdim\eql@marginleft@>\eql@flushleftmargin@
3099 \eql@marginleft@\eql@flushleftmargin@
3100 \else
3101 \ifdim\eql@marginleft@<\eql@flushleftmarginmin@
3102 \eql@marginleft@\eql@flushleftmarginmin@
3103 \fi
3104 \fi
3105 \else
3106 \eql@marginleft@\eql@flushleftmargin@
3107 \fi
3108 \else
3109 \ifdefined\eql@align@argins
3110 \eql@marginleft@\displaywidth
3111 \advance\eql@marginleft@-\eql@totalwidth@
3112 \advance\eql@marginleft@-\eql@align@inter@\eql@colsep@
3113 \ifdim\eql@marginleft@<\eql@tagmargin@
3114 \ifdim\ifdefined\eql@tagsleft\eql@marginleft@\else-\p@fi<z@
3115 \eql@marginleft@\z@
3116 \fi
3117 \else
3118 \advance\eql@marginleft@-\eql@tagmargin@
3119 \divide\eql@marginleft@\tw@
3120 \fi

```

```

3121 \else
3122 \eql@marginleft@z@
3123 \fi
3124 \fi

3125 \ifdefined\eql@tagsleft
3126 \let\eql@align@adjust@test\eql@align@adjust@test@tagleft
3127 \else
3128 \let\eql@align@adjust@test\eql@align@adjust@test@tagright
3129 \fi

3130 \loop\ifnum\eql@row@>z@
3131 \ifnum\eql@numbering@target@<z@
3132 \eql@tagwidth@\eql@tagwidth@get\eql@row@\relax
3133 \else
3134 \ifnum\eql@numbering@target@=\eql@row@
3135 \eql@tagwidth@\eql@tagwidth@getz@\relax
3136 \fi
3137 \fi
3138 \ifdim\eql@tagwidth@>z@
3139 \eql@align@adjust@width
3140 \eql@align@adjust@test
3141 \fi
3142 \advance\eql@row@\m@ne
3143 \repeat

3144 \advance\eql@totalwidth@\eql@align@inter@\eql@colsep@
3145 \advance\eql@totalwidth@\eql@marginleft@
3146 }

```

Calc Space. **TODO:** describe

```

3147 \def\eql@align@adjust@width{%
3148 \eql@line@pos@z@
3149 \eql@column@z@
3150 \eql@line@avail@\eql@totalwidth@
3151 \@tempcnta\eql@totalcolumns@
3152 \eql@line@width@z@
3153 \@tempcntb@z@
3154 \edef\@tempb{\eql@fieldlength@get\eql@row@}%
3155 \@for\@tempa:=\@tempb\do
3156 \eql@align@adjust@width@col
3157 \advance\@tempcnta\m@ne
3158 \divide\@tempcnta\tw@
3159 \advance\@tempcntb\m@ne
3160 \divide\@tempcntb\tw@
3161 }

```

TODO: describe

```

3162 \def\eql@align@adjust@width@col{%
3163 \advance\eql@column@\@ne
3164 \@tempdima\@tempa\relax
3165 \dimen@\eql@align@colwidth@get\eql@column@\relax
3166 \ifdim\@tempdima>z@
3167 \ifdim\eql@line@width@=\z@
3168 \eql@line@avail@\eql@line@pos@
3169 \@tempcnta\eql@column@
3170 \ifodd\eql@column@
3171 \advance\eql@line@avail@\dimen@

```

```

3172     \advance\eqlline@avail@-\@tempdima
3173     \fi
3174     \fi
3175     \eqlline@width@\eqlline@pos@
3176     \@tempcntb\eqlline@column@
3177     \ifodd\eqlline@column@
3178     \advance\eqlline@width@\dimen@
3179     \else
3180     \advance\eqlline@width@\@tempdima
3181     \fi
3182     \fi
3183     \advance\eqlline@pos@\dimen@
3184 }

```

Placement for right tags. **TODO:** describe

@align@adjust@test@r

```

3185 \def\eql@align@adjust@test@tagright{%
3186     \@tempdima\eqlline@width@
3187     \advance\@tempdima\eqlline@tagwidth@
3188     \dimen@\eqlline@marginleft@
3189     \advance\dimen@\@tempdima
3190     \advance\dimen@\@tempcntb\eqlline@colsep@
3191     \ifdim\dimen@>\displaywidth
3192     \eql@align@adjust@modify@tagright
3193     \fi
3194 }

```

TODO: describe

```

3195 \def\eql@align@adjust@modify@tagright{%
3196     \dimen@\eqlline@colsepmin@
3197     \multiply\dimen@\@tempcntb
3198     \advance\dimen@\@tempdima
3199     \ifdefined\eqlline@flushleft
3200     \advance\dimen@\eqlline@flushleftmarginmin@
3201     \fi
3202     \ifdim\dimen@>\displaywidth\else

```

TODO: check full length case? are the shifts sufficient in all possible cases?! is the left margin positive?!

```

3203     \ifnum\@tempcntb>\z@
3204     \dimen@\displaywidth
3205     \advance\dimen@-\@tempdima
3206     \count@\@tempcntb
3207     \ifdefined\eqlline@flushleft
3208     \advance\dimen@-\eqlline@marginleft@
3209     \else
3210     \ifdefined\eqlline@align@margin@
3211     \advance\count@\@ne
3212     \fi
3213     \fi
3214     \divide\dimen@\count@
3215     \ifdim\dimen@<\eqlline@colsep@
3216     \ifdim\dimen@<\eqlline@colsepmin@

```

```

3217         \dimen@eql@colsepmin@
3218     \fi
3219     \eql@colsep@\dimen@
3220 \fi
3221 \fi

```

TODO: could this possibly be less than the minimum margin? (we checked above, but under the assumption that `\eql@colsep@` is minimal!)

```

3222     \dimen@\displaywidth
3223     \advance\dimen@-\@tempdima
3224     \advance\dimen@-\@tempcntb\eql@colsep@
3225     \ifdim\dimen@<\eql@marginleft@
3226         \eql@marginleft@\dimen@
3227     \fi
3228 \fi
3229 }

```

Placement for left tags. TODO: describe

```

3230 \def\eql@align@adjust@test@tagleft{%
3231     \count@eql@align@inter@
3232     \advance\count@-\@tempcnta
3233     \@tempdima-\eql@line@avail@
3234     \advance\@tempdima\eql@tagwidth@
3235     \dimen@eql@marginleft@
3236     \advance\dimen@\@tempcnta\eql@colsep@
3237     \ifdim\dimen@<\@tempdima
3238         \eql@align@adjust@modify@tagleft
3239     \fi
3240 }

```

```

3241 \def\eql@align@adjust@modify@tagleft{%

```

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

```

3242     \ifdefined\eql@flushleft
3243         \dimen@eql@flushleftmarginmax@
3244     \else
3245         \dimen@\displaywidth
3246     \fi
3247     \ifdim\dimen@>\eql@tagwidth@

3248     \dimen@eql@colsepmin@
3249     \multiply\dimen@\count@
3250     \advance\dimen@\eql@totalwidth@
3251     \advance\dimen@\@tempdima
3252     \ifdim\dimen@>\displaywidth\else

3253         \ifnum\count@>0
3254             \dimen@\displaywidth
3255             \advance\dimen@-\eql@totalwidth@
3256             \advance\dimen@-\@tempdima
3257             \ifdefined\eql@align@margin@
3258                 \advance\count@\@ne
3259             \fi
3260             \divide\dimen@\count@
3261             \ifdim\dimen@<\eql@colsep@

```

```

3262         \ifdim\dimen@<\eql@colsepmin@
3263         \dimen@\eql@colsepmin@
3264         \fi
3265         \advance\dimen@-\eql@colsep@
3266         \advance\eql@marginleft@-\eql@align@inter@\dimen@
3267         \advance\eql@colsep@\dimen@
3268     \fi
3269 \fi

3270 \dimen@-\@tempcnta\eql@colsep@
3271 \advance\dimen@\@tempdima
3272 \ifdim\dimen@>\eql@marginleft@
3273     \eql@marginleft@\dimen@
3274 \fi
3275 \fi
3276 \fi
3277 }

```

P Interface

P.1 Scanning the Environment’s Body

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

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

```

3278 \newtoks\eql@scan@reg@

TODO: describe

3279 \def\eql@scan@body@dump{\the\eql@scan@reg@}
3280 \def\eql@scan@body@rescan{%
3281     \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
3282 \let\eql@scan@body\eql@scan@body@dump

```

`\eql@scan@addto` Then we define a macro to add something (i.e. its argument) to the token register `\@envbody`:

```

3283 \def\eql@scan@addto#1{\eql@scan@reg@\expandafter{\the\eql@scan@reg@#1}}

```

`\eql@scan@env` The macro `\eql@scan@env` starts the scan for the `\end{...}` command of the current environment. It takes a macro name as argument. This macro is supposed to take the whole body of the environment as its argument:

```

3284 \def\eql@scan@env#1{%
3285     (dev)\eql@dev@enter\eql@scan@env
3286     \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenenv}}%
3287     \eql@scan@reg@{\}\def\eql@scan@stack{b}%

```

If we simply called `\eql@scan@env@iterate` directly, the error message for a `\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`:

```

3288 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
3289 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
3290 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
3291 \eql@scan@iterate
3292 }

```

`\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.

```

3293 \def\eql@scan@env@count#1\begin#2{%
3294 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
3295 }

```

`\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 by the `\push@begins` function. 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:

```

3296 \def\eql@scan@env@iterate#1\end#2{%
3297 \edef\eql@scan@stack{%
3298 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
3299 \ifx\@empty\eql@scan@stack
3300 \@checkend{#2}%
3301 \eql@scan@addto{#1}%
3302 \expandafter\let\eql@scan@iterate\eql@scan@env@org
3303 (dev)\eql@dev@leave\eql@scan@env
3304 \expandafter\eql@scan@end
3305 \else
3306 \eql@scan@addto{#1\end{#2}}%
3307 \expandafter\eql@scan@iterate
3308 \fi
3309 }

```

TODO: describe

```

3310 \def\eql@scan@env@cancel{%
3311 \@namedef{end\@currenvir}{\ignorespacesafterend}%
3312 }

```

square brackets **TODO:** describe

```

3313 \def\eql@scan@sqr#1{%
3314 (dev)\eql@dev@enter\eql@scan@sqr
3315 \def\eql@scan@end{#1\]}%
3316 \eql@scan@reg@{\def\eql@scan@stack{b}%
3317 \let\eql@scan@sqr@org\[%\]
3318 \let\[\eql@scan@sqr@iterate%\]
3319 \[%\]
3320 }

```

TODO: describe

```

3321 \def\eql@scan@sqr@count#1\[#2{\%]
3322 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
3323 }

```

TODO: describe

```

3324 \def\eql@scan@sqr@iterate#1\]{%
3325   \edef\eql@scan@stack{%
3326     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
3327   \ifx\@empty\eql@scan@stack
3328     \let\[\eql@scan@sqr@org%\]
3329     \eql@scan@addto{#1}%
3330 (dev)\eql@dev@leave\eql@scan@sqr
3331   \expandafter\eql@scan@end
3332   \else
3333     \eql@scan@addto{#1\]}%
3334   \expandafter\[\]
3335   \fi
3336 }

3337 \def\eql@scan@sqrang@cancel{\expandafter\ignorespaces\@gobble}

```

anuglar brackets **TODO:** describe

```

3338 \def\eql@scan@ang#1{%
3339 (dev)\eql@dev@enter\eql@scan@ang
3340   \def\eql@scan@end{#1\>}%
3341   \eql@scan@reg@{\}\def\eql@scan@stack{b}%
3342   \let\eql@scan@ang@org\<%\>
3343   \let\<\eql@scan@ang@iterate%\>
3344   \<%\>
3345 }

```

TODO: describe

```

3346 \def\eql@scan@ang@count#1\<#2%\>
3347   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
3348 }

```

TODO: describe

```

3349 \def\eql@scan@ang@iterate#1\>{%
3350   \edef\eql@scan@stack{%
3351     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
3352   \ifx\@empty\eql@scan@stack
3353     \let\<\eql@scan@ang@org%\>
3354     \eql@scan@addto{#1}%
3355 (dev)\eql@dev@leave\eql@scan@ang
3356   \expandafter\eql@scan@end
3357   \else
3358     \eql@scan@addto{#1\>}%
3359   \expandafter\<%\>
3360   \fi
3361 }

```

P.2 Options Processing

```

3362 \def\eql@equations@testall{\eql@equations@testtilde}
3363 \def\eql@equations@testtilde#1{%
3364   \eql@ifnextgobble@tight~%
3365   {\eqnadopt{lines}\eql@equations@testopt{#1}}%
3366   {\eql@equations@testopt{#1}}}
3367 \def\eql@equations@testopt#1{%
3368   \eql@ifnextchar@tight[%]
3369   {\eql@equations@adopt{\eql@equations@testexcl{#1}}}%

```

```

3370   {\eql@equations@testexcl{#1}}
3371 \def\eql@equations@adopt#1[#2]{\eqnadopt{#2}#1}
3372 \def\eql@equations@testexcl#1{%
3373   \eql@ifnextgobble@tight!%
3374   {\eqnadopt{donumber}\eql@equations@testat{#1}}%
3375   {\eql@equations@teststar{#1}}}
3376 \def\eql@equations@teststar#1{%
3377   \eql@ifstar@tight%
3378   {\eqnadopt{nonumber}\eql@equations@testat{#1}}%
3379   {\eql@equations@testat{#1}}}
3380 \def\eql@equations@testat#1{%
3381   \eql@ifat@tight
3382   {\eql@equations@addlabel{#1}}%
3383   {#1}}
3384 \def\eql@equations@addlabel#1#2{\eqnadopt{label={#2}}#1}
3385 \def\eql@equations@processopt{%
3386   \let\eql@blocklabel\@undefined
3387   \let\eql@blocktag\@undefined
3388   \let\eql@skip@force@above\@undefined
3389   \let\eql@skip@force@below\@undefined
3390   \let\eql@skip@force@leave\@undefined
3391   \eql@abovespace@\z@skip
3392   \eql@belowspace@\z@skip
3393   \eql@displaybreak@prepen@\@MM
3394   \eql@nextopt@process{equations}%
3395   \let\eql@punct@block\eql@punct@main
3396   \let\eql@punct@main\relax
3397   \eql@indent@\glueexpr\eql@indent@val\relax
3398 }

```

P.3 Environments

equations (*env.*)

```

3399 \newenvironment{equations}{%
3400 (dev)\eql@dev@enterenv
3401   \ifmmode
3402     \eql@error@mathmode{\string\begin{\@currenvir}}%
3403     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
3404   \else
3405     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3406     \expandafter\eql@equations@start
3407   \fi
3408 }{%
3409   \eql@equations@end
3410   \ignorespacesafterend
3411 (dev)\eql@dev@leaveenv
3412 }

3413 \def\eql@equations@start{%
3414   \eql@equations@processopt
3415   \ifdefined\eql@equations@main
3416     \expandafter\eql@scan@env\expandafter\eql@equations@main
3417   \else
3418     \expandafter\eql@single@start
3419   \fi
3420 }

```

square and angle brackets

```
3421 \newenvironment{equations@sqr}{\{}}
3422 \def\eqlequations@sqr@opt{equation,nonumber}
3423 \protected\def\eqlequations@sqr@open{%
3424   \ifmmode
3425     \eqlequations@mathmode{\string\[\dots\string\]}%
3426     \expandafter\eqlequations@scan@sqr\expandafter\eqlequations@scan@sqrang@cancel
3427   \else
3428 (dev)\eqlequations@dev@enter{\[\dots\string\]}%
3429   \expandafter\eqlequations@eqnaddopt\expandafter{\eqlequations@sqr@opt}%
3430   \begin{equations@sqr}%
3431   \expandafter\eqlequations@ampprotect\expandafter\eqlequations@testall
3432   \expandafter\eqlequations@sqr@start
3433   \fi
3434 }

3435 \newenvironment{equations@ang}{\{}}
3436 \def\eqlequations@ang@opt{align,nonumber}
3437 \protected\def\eqlequations@ang@open{%
3438   \ifmmode
3439     \eqlequations@mathmode{\string\langle\dots\string\rangle}%
3440     \expandafter\eqlequations@scan@ang\expandafter\eqlequations@scan@sqrang@cancel
3441   \else
3442 (dev)\eqlequations@dev@enter{\langle\dots\string\rangle}%
3443   \expandafter\eqlequations@eqnaddopt\expandafter{\eqlequations@ang@opt}%
3444   \begin{equations@ang}%
3445   \expandafter\eqlequations@ampprotect\expandafter\eqlequations@testall
3446   \expandafter\eqlequations@ang@start
3447   \fi
3448 }

3449 \def\eqlequations@sqr@start{%
3450   \eqlequations@processopt
3451   \ifdefined\eqlequations@main
3452     \expandafter\eqlequations@scan@sqr\expandafter\eqlequations@main
3453   \else
3454     \expandafter\eqlequations@single@start
3455   \fi
3456 }

3457 \def\eqlequations@ang@start{%
3458   \eqlequations@processopt
3459   \ifdefined\eqlequations@main
3460     \expandafter\eqlequations@scan@ang\expandafter\eqlequations@main
3461   \else
3462     \expandafter\eqlequations@single@start
3463   \fi
3464 }

3465 \protected\def\eqlequations@sqr@close{%
3466   \eqlequations@end
3467 (dev)\eqlequations@dev@leave{\[\dots\string\]}%
3468   \end{equations@sqr}%
3469   \ignorespaces
3470 }

3471 \protected\def\eqlequations@ang@close{%
3472   \eqlequations@end
3473 (dev)\eqlequations@dev@leave{\langle\dots\string\rangle}%
```

```

3474 \end{equations@ang}%
3475 \ignorespaces
3476 }

```

P.4 Independent Routines of amsmath

`\eqref` amsmath defines the macro `\eqref` to refer to equation labels in a proper format We provide it for completeness: **TODO**: versions of `eqref` for ranges (optional argument) or lists (,)?

```

3477 \DeclareRobustCommand{\eql@eqref}[1]{\textup{\eql@tag@boxedform{\ref{#1}}}}

```

`\numberwithin` amsmath defines the macro `\numberwithin` to specify that equations should be numbered within sections. The L^AT_EX kernel contains a similar command `\counterwithin` (with a slightly extended syntax) that can be used as a drop-in replacement for `\numberwithin`:

```

3478 \DeclareRobustCommand{\eql@numberwithin}[3][\arabic]{%
3479 \@ifundefined{c@#2}{\@nocounterr{#2}}{%
3480 \@ifundefined{c@#3}{\@nocounterr{#3}}{%
3481 \ifdefined\HyOrg@addtoreset
3482 \HyOrg@addtoreset{#2}{#3}%
3483 \else
3484 \@addtoreset{#2}{#3}%
3485 \fi
3486 \expandafter\xdef\csname the#2\endcsname{%
3487 \expandafter\noexpand\csname the#3\endcsname.\noexpand#1{#2}}%
3488 \ifdefined\theHequation
3489 \expandafter\xdef\csname theH#2\endcsname{%
3490 \expandafter\noexpand
3491 \csname the\@ifundefined{theH#3}{H#3\endcsname.%
3492 \noexpand\arabic{#2}}%
3493 \fi
3494 }%
3495 }%
3496 }

```

`\allowdisplaybreaks` amsmath defines the macro `\allowdisplaybreaks` which we also provide for completeness. The package uses the general setup mechanism instead:

```

3497 \DeclareRobustCommand{\eql@allowdisplaybreaks}[1][4]{%
3498 \eqnlineset{allowbreaks=#1}%
3499 }

```

`\notag` amsmath defines the alias `\notag` for `\nonumber` which we also provide for completeness:

```

3500 \newcommand{\eql@notag}{\nonumber}

```

`\thetag` amsmath defines `\thetag` to print a tag which we also provide for completeness:

```

3501 \newcommand{\eql@thetag}{\leavevmode\eql@tag@boxedform}

```

Q Options

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

```

3502 \RequirePackage{keyval}

```

Q.1 Selection Tools

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

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

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

```

3503 \def\eql@decide@select#1#2#3#4{%
3504   \def\@tempa{#3}%
3505   \let\@tempd\@undefined
3506   \@for\@tempc:=#4\do{%
3507     \ifdefined\@tempd\else
3508       \edef\@tempb{\noexpand\@tempb:=\expandafter\@firstoftwo\@tempc}%
3509       \expandafter\@for\@tempb\do{%
3510         \ifx\@tempa\@tempb
3511           \expandafter\expandafter\expandafter\def
3512             \expandafter\expandafter\expandafter\@tempd
3513               \expandafter\expandafter\expandafter{%
3514                 \expandafter\@secondoftwo\@tempc}%
3515         \fi
3516       }%
3517     \fi
3518   }%
3519   \ifdefined\@tempd
3520     \@tempd
3521   \else
3522     \eql@error{undefined value ‘#3’ for option ‘#2’ of ‘#1’}%
3523   \fi
3524 }
```

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

```

3525 \def\eql@decide@if#1#2#3#4#5{%
3526   \eql@decide@select{#1}{#2}{#3}{%
3527     {\on,true,yes,enabled,1}{#4}},%
3528     {\off,false,no,disabled,0}{#5}}}
```

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

```

3529 \def\eql@decide@bool#1#2#3#4{%
3530   \eql@decide@if{#1}{#2}{#3}{\let#4\eql@true}{\let#4\eql@false}}
```

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

```

3531 \def\eql@decide@abovebelow#1#2#3#4#5{%
3532   \eql@decide@select{#1}{#2}{#3}{%
3533     {,abovebelow,both,tb}{#4#5},%
3534     {above,top,t}{#4},%
3535     {below,bottom,b}{#5}}%
3536 }
```

Q.2 Declaration Code

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

```

3537 \def\eql@define@key#1#2{%
3538   \eql@ifnextchar@loose[%]
3539     {\eql@definekey@opt{#1}{#2}}%
3540     {\eql@definekey@noopt{#1}{#2}}%
3541 }
3542 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{#3}}
3543 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{#3}{#4}}
3544 \def\eql@definekey@for#1#2#3{%
3545   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}##3}%
3546   \edef\eql@for@vara{\noexpand\eql@for@vara:=#1}%
3547   \expandafter\@for\eql@for@vara\do{%
3548     \edef\eql@for@varb{\noexpand\eql@for@varb:=#2}%
3549     \expandafter\@for\eql@for@varb\do{%
3550       \edef\eql@for@call##1{%
3551         \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@vara}}%
3552       \eql@for@call{##1}}%
3553     }%
3554   }%
3555 }
```

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

```

3556 \def\eql@setkeys#1#2{%
3557   \def\eql@tmp{\setkeys{eql@#1}}%
3558   \expandafter\eql@tmp\expandafter{#2}%
3559 }
```

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

```

\eql@nextopt
\eqnaddopt
3560 \let\eql@nextopt\@empty
3561 \def\eql@nextopt@process#1{%
3562 (dev)\eql@dev@start\eql@nextopt@process
3563   \eql@setkeys{#1}\eql@nextopt
3564   \let\eql@tagging@opt\eql@nextopt
3565   \global\let\eql@nextopt\@empty
3566 }
3567 \newcommand{\eqnaddopt}[1]{%
3568   \expandafter\def\expandafter\eql@nextopt\expandafter{\eql@nextopt,#1}}
```

Q.3 Parameter Sets

TODO: : `par@above` plus `parskip`?

```

3569 \def\eql@defaults@classic{%
3570   \def\eql@colsepmax@val{.5\maxdimen}%
3571   \def\eql@spread{\jot}%
3572   \let\eql@display@height\@undefined
3573   \let\eql@display@depth\@undefined
3574   \eql@skip@mode@short@tw@
3575   \eql@skip@mode@cont@\@ne
```

```

3576 \eql@skip@mode@par@z@
3577 \eql@skip@mode@top@z@
3578 \def\eql@skip@long@above{\abovedisplayskip}%
3579 \def\eql@skip@long@below{\belowdisplayskip}%
3580 \def\eql@skip@short@above{\abovedisplayshortskip}%
3581 \def\eql@skip@short@below{\belowdisplayshortskip}%
3582 \def\eql@skip@cont@above{\eql@skip@short@above}%
3583 \def\eql@skip@cont@below{\eql@skip@short@below}%
3584 \def\eql@skip@par@above{\eql@skip@long@above}%
3585 \def\eql@skip@par@below{\eql@skip@long@below}%
3586 \def\eql@skip@top@above{\eql@skip@long@above}%
3587 \def\eql@skip@top@below{\eql@skip@long@below}%
3588 \def\eql@skip@med@above{\abovedisplayskip/2}%
3589 \def\eql@skip@med@below{\belowdisplayskip/2}%
3590 \def\eql@skip@medtag@above{\z@skip}%
3591 \def\eql@skip@medtag@below{\z@skip}%
3592 \def\eql@skip@tag@above{\z@skip}%
3593 \def\eql@skip@tag@below{\z@skip}%
3594 }

```

values based on 10pt vs 12pt

```

3595 \def\eql@defaults@eqnlines{%
3596 \def\eql@colsepmax@val{2em}%
3597 \def\eql@spread{0.2\normalbaselineskip}%
3598 \def\eql@display@height{\ht\eql@strutbox}%
3599 \def\eql@display@depth{\dp\eql@strutbox}%
3600 \eql@skip@mode@short@one
3601 \eql@skip@mode@cont@z@
3602 \eql@skip@mode@par@z@
3603 \eql@skip@mode@top@z@
3604 \def\eql@skip@long@above{0.75\normalbaselineskip
3605 \@plus0.25\normalbaselineskip\@minus0.4\normalbaselineskip}%
3606 \let\eql@skip@long@below\eql@skip@long@above
3607 \def\eql@skip@short@above{0.0\normalbaselineskip
3608 \@plus0.2\normalbaselineskip}%
3609 \def\eql@skip@short@below{0.0\normalbaselineskip
3610 \@plus0.25\normalbaselineskip}%
3611 \def\eql@skip@med@above{0.375\normalbaselineskip
3612 \@plus0.225\normalbaselineskip\@minus0.2\normalbaselineskip}%
3613 \let\eql@skip@med@below\eql@skip@med@above
3614 \def\eql@skip@cont@above{\z@skip}%
3615 \def\eql@skip@cont@below{\eql@skip@long@below}%
3616 \def\eql@skip@par@above{\eql@skip@long@above}%
3617 \def\eql@skip@par@below{\eql@skip@long@below}%
3618 \def\eql@skip@top@above{\z@skip}%
3619 \def\eql@skip@top@below{\z@skip}%
3620 \def\eql@skip@tag@above{\z@skip}%
3621 \def\eql@skip@tag@below{\z@skip}%
3622 \def\eql@skip@partag@above{\z@skip}%
3623 \def\eql@skip@partag@below{\z@skip}%
3624 \def\eql@skip@medtag@above{\z@skip}%
3625 \def\eql@skip@medtag@below{\z@skip}%
3626 }
3627 \eql@defaults@eqnlines

```

Q.4 Options Declarations

TODO: describe

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

```
3628 \eql@define@key{equationsbox}{gathered,gather,ga,lines,lined,ln,\string~}[]{%
3629   \eql@mode@lined}
3630 \eql@define@key{equationsbox}{aligned,align,al,columns,col,@}[]{%
3631   \eql@mode@aligned}
3632 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
3633 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
3634 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\ vbox}
3635 \eql@define@key{equationsbox}{colsep}{\def\eql@box@colsep{#1}}
```

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

```
3636 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@equation}
3637 \eql@define@key{equations}{gathered,gather,ga,lines,lined,ln,\string~}[]{%
3638   \eql@mode@lines}
3639 \eql@define@key{equations}{aligned,align,al,columns,col,@}[]{%
3640   \eql@mode@align}
3641 \eql@define@key{equations}{native}[true]{%
3642   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
3643   \ifdefined\eql@single@native\let\eql@flushleft\eql@false\fi}
3644 \eql@define@key{setup}{native}[true]{%
3645   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
3646 \eql@define@key{setup}{scanequation}[true]{%
3647   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscans}
3648 \eql@define@key{setup}{sqropt}[]{%
3649   \def\eql@equations@sqr@opt{equation,#1}}
3650 \eql@define@key{setup}{angopt}[]{%
3651   \def\eql@equations@ang@opt{align,#1}}
```

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

```
3652 \def\eql@keycat{equations,equationsbox,setup}
3653 \eql@define@key\eql@keycat{spread}\jot{\def\eql@spread{#1}}
3654 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@if{#3}{#2}{#1}%
3655   {\let\eql@strut@field\eql@strut}{\let\eql@strut@field\relax}}
3656 \eql@define@key\eql@keycat{strutttag}[true]{\eql@decide@if{#3}{#2}{#1}%
3657   {\let\eql@strut@tag\eql@strut}{\let\eql@strut@tag\relax}}
```

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

```
3658 \eql@define@key{equations}{displaybreak}[4]{\eql@displaybreak@pre{#1}}
3659 \def\eql@keycat{equations,setup}
3660 \eql@define@key\eql@keycat{allowbreaks,allowdisplaybreaks}[4]{%
3661   \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax}
3662 \eql@define@key\eql@keycat{displayheight}[\ht\eql@strutbox@]{%
3663   \def\eql@display@height{#1}}
3664 \eql@define@key\eql@keycat{displaydepth}[\dp\eql@strutbox@]{%
3665   \def\eql@display@depth{#1}}
3666 \eql@define@key\eql@keycat{displayheight*}[]{%
3667   \let\eql@display@height\undefined}
3668 \eql@define@key\eql@keycat{displaydepth*}[]{%
3669   \let\eql@display@depth\undefined}
```

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

```

3670 \eql@define@key{equations}{noskip}[]{%
3671   \eql@decide@abovebelow{#3}{#2}{#1}%
3672   {\def\eql@skip@force@above{5}}%
3673   {\def\eql@skip@force@below{5}}%
3674 \eql@define@key{equations}{short}[above]{%
3675   \eql@decide@abovebelow{#3}{#2}{#1}%
3676   {\def\eql@skip@force@above{1}}%
3677   {\def\eql@skip@force@below{1}}%
3678 \eql@define@key{equations}{long}[]{%
3679   \eql@decide@abovebelow{#3}{#2}{#1}%
3680   {\def\eql@skip@force@above{0}}%
3681   {\def\eql@skip@force@below{0}}%
3682 \eql@define@key{equations}{medskip}[]{%
3683   \eql@decide@abovebelow{#3}{#2}{#1}%
3684   {\def\eql@skip@force@above{6}}%
3685   {\def\eql@skip@force@below{6}}%
3686 \eql@define@key{equations}{par}[par]{%
3687   \eql@decide@select{#3}{#2}{#1}{%
3688     {\default,}\let\eql@skip@force@leave\undefined},%
3689     {\cont,hmode}\let\eql@skip@force@leave\z@},%
3690     {\par,vmode}\let\eql@skip@force@leave\ne
3691     \ifdefined\eql@skip@force@below\else
3692     \def\eql@skip@force@below{3}%
3693     \fi},%
3694     {\top}\let\eql@skip@force@leave\tw@
3695     \ifdefined\eql@skip@force@below\else
3696     \def\eql@skip@force@below{4}
3697     \fi}}}}

```

TODO: describe

```

3698 \eql@define@key{equations}{skip}{%
3699   \def\eql@skip@force@above{7}%
3700   \def\eql@skip@custom@above{#1}%
3701   \let\eql@skip@force@below\eql@skip@force@above
3702   \let\eql@skip@custom@below\eql@skip@custom@above}
3703 \eql@define@key{equations}{aboveskip}{%
3704   \def\eql@skip@force@above{7}%
3705   \def\eql@skip@custom@above{#1}}
3706 \eql@define@key{equations}{belowskip}{%
3707   \def\eql@skip@force@below{7}%
3708   \def\eql@skip@custom@below{#1}}
3709 \eql@define@key{equations}{abovespace}{%
3710   \advance\eql@abovespace@glueexpr#1\relax}
3711 \eql@define@key{equations}{belowspace}{%
3712   \advance\eql@belowspace@glueexpr#1\relax}

```

TODO: describe

```

3713 \eql@define@key{intertext}{skip}{%
3714   \def\eql@skip@force@above{7}%
3715   \def\eql@skip@custom@above{#1}%
3716   \let\eql@skip@force@below\eql@skip@force@above
3717   \let\eql@skip@custom@below\eql@skip@custom@above}
3718 \eql@define@key{intertext}{aboveskip}{%
3719   \def\eql@skip@force@below{7}%
3720   \def\eql@skip@custom@below{#1}}

```

```

3721 \eql@define@key{intertext}{belowskip}{%
3722   \def\eql@skip@force@above{7}%
3723   \def\eql@skip@custom@above{#1}}
3724 \eql@define@key{intertext}{noskip}[]{%
3725   \eql@decide@abovebelow{#3}{#2}{#1}%
3726   {\def\eql@skip@force@below{5}}%
3727   {\def\eql@skip@force@above{5}}}
3728 \eql@define@key{intertext}{short}[]{%
3729   \eql@decide@abovebelow{#3}{#2}{#1}%
3730   {\def\eql@skip@force@below{1}}%
3731   {\def\eql@skip@force@above{1}}}
3732 \eql@define@key{intertext}{long}[]{%
3733   \eql@decide@abovebelow{#3}{#2}{#1}%
3734   {\def\eql@skip@force@below{0}}%
3735   {\def\eql@skip@force@above{0}}}
3736 \eql@define@key{intertext}{medskip}[]{%
3737   \eql@decide@abovebelow{#3}{#2}{#1}%
3738   {\def\eql@skip@force@below{6}}%
3739   {\def\eql@skip@force@above{6}}}

```

TODO: describe

```

3740 \eql@define@key{setup}{skip, longskip}{%
3741   \abovedisplayskip\glueexpr#1\relax
3742   \belowdisplayskip\abovedisplayskip
3743   \def\eql@skip@long@above{#1}%
3744   \let\eql@skip@long@below\eql@skip@long@above}
3745 \eql@define@key{setup}{aboveskip, abovelongskip}{%
3746   \abovedisplayskip\glueexpr#1\relax
3747   \def\eql@skip@long@above{#1}}
3748 \eql@define@key{setup}{belowskip, belowlongskip}{%
3749   \belowdisplayskip\glueexpr#1\relax
3750   \def\eql@skip@long@below{#1}}
3751 \eql@define@key{setup}{aboveshortskip}{%
3752   \abovedisplayshortskip\glueexpr#1\relax
3753   \def\eql@skip@short@above{#1}}
3754 \eql@define@key{setup}{belowshortskip}{%
3755   \belowdisplayshortskip\glueexpr#1\relax
3756   \def\eql@skip@short@below{#1}}
3757 \eql@define@key{setup}{tagskip}{%
3758   \def\eql@skip@tag@above{#1}%
3759   \let\eql@skip@tag@below\eql@skip@tag@above}
3760 \eql@define@key{setup}{abovetagskip}{%
3761   \def\eql@skip@tag@above{#1}}
3762 \eql@define@key{setup}{belowtagskip}{%
3763   \def\eql@skip@tag@below{#1}}
3764 \eql@define@key{setup}{medskip}{%
3765   \def\eql@skip@med@above{#1}%
3766   \let\eql@skip@med@below\eql@skip@med@above}
3767 \eql@define@key{setup}{abovemedskip}{%
3768   \def\eql@skip@med@above{#1}}
3769 \eql@define@key{setup}{belowmedskip}{%
3770   \def\eql@skip@med@below{#1}}
3771 \eql@define@key{setup}{medtagskip}{%
3772   \def\eql@skip@medtag@above{#1}%
3773   \let\eql@skip@medtag@below\eql@skip@medtag@above}
3774 \eql@define@key{setup}{abovemedtagskip}{%
3775   \def\eql@skip@medtag@above{#1}}
3776 \eql@define@key{setup}{belowmedtagskip}{%

```

```

3777 \def\eql@skip@medtag@below{#1}}
3778 \eql@define@key{setup}{abovetopskip}{%
3779 \def\eql@skip@top@above{#1}}
3780 \eql@define@key{setup}{belowtopskip}{%
3781 \def\eql@skip@top@below{#1}}
3782 \eql@define@key{setup}{aboveparskip}{%
3783 \def\eql@skip@par@above{#1}}
3784 \eql@define@key{setup}{belowparskip}{%
3785 \def\eql@skip@par@below{#1}}
3786 \eql@define@key{setup}{abovepartagskip}{%
3787 \def\eql@skip@partag@above{#1}}
3788 \eql@define@key{setup}{belowpartagskip}{%
3789 \def\eql@skip@partag@below{#1}}
3790 \eql@define@key{setup}{abovecontskip}{%
3791 \def\eql@skip@cont@above{#1}}
3792 \eql@define@key{setup}{abovecontskip*}[]{%
3793 \def\eql@skip@cont@above{\eql@spread-\eql@skip@long@below}}
3794 \eql@define@key{setup}{belowcontskip}{%
3795 \def\eql@skip@cont@below{#1}}
3796 \eql@define@key{setup}{shortmode}{%
3797 \eql@decide@select{#3}{#2}{#1}{%
3798   {{off,never,no}{\eql@skip@mode@short@z}},%
3799   {{above,neverbelow,notbelow,belowoff}{\eql@skip@mode@short@one}},%
3800   {{belowone,belowsingle}{\eql@skip@mode@short@tw}},%
3801   {{belowall,always,on}{\eql@skip@mode@short@thr@}}}}
3802 \eql@define@key{setup}{belowcontmode}{%
3803 \eql@decide@select{#3}{#2}{#1}{%
3804   {{long}{\eql@skip@mode@cont@z}},%
3805   {{short}{\eql@skip@mode@cont@one}},%
3806   {{cont}{\eql@skip@mode@cont@tw@}}}}
3807 \eql@define@key{setup}{belowparmode}{%
3808 \eql@decide@select{#3}{#2}{#1}{%
3809   {{long}{\eql@skip@mode@par@z}},%
3810   {{short}{\eql@skip@mode@par@one}},%
3811   {{cont}{\eql@skip@mode@par@tw@}},%
3812   {{par}{\eql@skip@mode@par@thr@}}}}
3813 \eql@define@key{setup}{belowtopmode}{%
3814 \eql@decide@select{#3}{#2}{#1}{%
3815   {{long}{\eql@skip@mode@top@z}},%
3816   {{short}{\eql@skip@mode@top@one}},%
3817   {{cont}{\eql@skip@mode@top@tw@}},%
3818   {{par}{\eql@skip@mode@top@thre@}},%
3819   {{top}{\eql@skip@mode@top@4\relax}}}}

```

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

```

3820 \def\eql@keycat{equations,subequations}
3821 \eql@define@key\eql@keycat{label}{\eql@blocklabel@set{#1}}
3822 \eql@define@key\eql@keycat{tag}{\eql@blocktag@set{#1}}
3823 \eql@define@key\eql@keycat{tag*}{\eql@blocktag@setstar{#1}}

```

Tag Spacing. **TODO:** describe

```

3824 \def\eql@keycat{equations,setup}
3825 \eql@define@key\eql@keycat{tagmargin}{\eql@tagmargin@glueexpr#1\relax}
3826 \eql@define@key\eql@keycat{tagmargin*}{\settowidth\eql@tagmargin@{#1}}
3827 \eql@define@key\eql@keycat{mintagsep}{\eql@tagsepmin@glueexpr#1\relax}
3828 \eql@define@key\eql@keycat{mintagwidth}{\eql@tagwidthmin@glueexpr#1\relax}

```

3829 `\eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin@{#1}}`

Tag Layout. **TODO:** describe

3830 `\eql@define@key{setup}{taglayout}{\eql@tag@setbox{#1}}`
 3831 `\eql@define@key{setup}{taglayout*}{\eql@tag@setbox@{#1}}`
 3832 `\eql@define@key{setup}{tagform}{\eql@tag@setform#1}`
 3833 `\eql@define@key{setup}{tagform*}{\eql@tag@setform@{#1}}`
 3834 `\eql@define@key{setup}{subeqtemplate}{\def\eql@subequations@template{#1}}`
 3835 `\eql@define@key{setup}{autolabel}[true]{%`
 3836 `\eql@decide@bool{#3}{#2}{#1}\eql@autolabel}`
 3837 `\eql@define@key{setup}{autotag}[true]{%`
 3838 `\eql@decide@bool{#3}{#2}{#1}\eql@autotag}`

Equation Numbering. **TODO:** describe

3839 `\def\eql@keycat{equations,setup}`
 3840 `\eql@define@key\eql@keycat{numberline,numline,n}[all]{%`
 3841 `\eql@numbering@set{##1}}`
 3842 `\eql@define@key\eql@keycat{nonumber,nn,*}[]{\let\eql@numbering@active\eql@false}`
 3843 `\eql@define@key\eql@keycat{donumber,dn,!}[]{\let\eql@numbering@active\eql@true}`
 3844 `\eql@define@key\eql@keycat{number,num}[true]{%`
 3845 `\eql@decide@bool{#3}{#2}{#1}\eql@numbering@active}`
 3846 `\eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}`
 3847 `\eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}`
 3848 `\eql@define@key\eql@keycat{tags,eqno}{%`
 3849 `\eql@decide@select{#3}{#2}{#1}{%`
 3850 `{\right,r}{\let\eql@tagsleft\eql@false}},%`
 3851 `{\left,l}{\let\eql@tagsleft\eql@true}}}`

Horizontal Layout. **TODO:** describe

3852 `\def\eql@keycat{equations,setup}`
 3853 `\eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%`
 3854 `{\center,c}{\let\eql@flushleft\eql@false}},%`
 3855 `{\left,l}{\let\eql@flushleft\eql@true}}}`
 3856 `\eql@define@key\eql@keycat{center}[]{\let\eql@flushleft\eql@false}`
 3857 `\eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@flushleft\eql@true}`
 3858 `\eql@define@key\eql@keycat{leftmargin}{%`
 3859 `\eql@flushleftmargin@\glueexpr#1\relax}`
 3860 `\eql@define@key\eql@keycat{minleftmargin}{%`
 3861 `\eql@flushleftmarginmin@\glueexpr#1\relax}`
 3862 `\eql@define@key\eql@keycat{maxleftmargin}{%`
 3863 `\eql@flushleftmarginmax@\glueexpr#1\relax}`
 3864 `\eql@define@key\eql@keycat{maxleftmargin*}[]{%`
 3865 `\eql@flushleftmarginmax@.5\maxdimen}`

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

3866 `\def\eql@keycat{equations,setup}`
 3867 `\eql@define@key\eql@keycat{marginbadness}{\eql@marginbadness@#1\relax}`
 3868 `\eql@define@key\eql@keycat{maxbadness}{\eql@maxbadness@#1\relax}`
 3869 `\eql@define@key\eql@keycat{mincolsep}{\def\eql@colsepmin@val{#1}}`
 3870 `\eql@define@key\eql@keycat{maxcolsep}{\def\eql@colsepmax@val{#1}}`
 3871 `\eql@define@key\eql@keycat{maxcolsep*}[]{\def\eql@colsepmax@val{.5\maxdimen}}`
 3872 `\eql@define@key\eql@keycat{margins}[true]{%`
 3873 `\eql@decide@bool{#3}{#2}{#1}\eql@align@margins}`

```

3874 \def\eq@keycat{equationsbox,setup}
3875 \eq@define@key\eq@keycat{margin}{%
3876   \def\eq@box@marginleft{#1}\def\eq@box@marginright{#1}}
3877 \eq@define@key\eq@keycat{marginleft}{\def\eq@box@marginleft{#1}}
3878 \eq@define@key\eq@keycat{marginright}{\def\eq@box@marginright{#1}}

```

Horizontal Shape. **TODO:** describe

```

3879 \def\eq@keycat{equations, equationsbox, setup}
3880 \eq@define@key\eq@keycat{shape}[default]{\eq@shape@set{#1}}
3881 \eq@define@key\eq@keycat{padding, pad}[\eq@indent@val]{%
3882   \let\eq@paddingmax\eq@false
3883   \def\eq@paddingleft@val{#1}\def\eq@paddingright@val{#1}}
3884 \eq@define@key\eq@keycat{padleft}[\eq@indent@val]{%
3885   \let\eq@paddingmax\eq@false\def\eq@paddingleft@val{#1}}
3886 \eq@define@key\eq@keycat{padright}[\eq@indent@val]{%
3887   \let\eq@paddingmax\eq@false\def\eq@paddingright@val{#1}}
3888 \eq@define@key\eq@keycat{padmax}[true]{%
3889   \eq@decide@bool{#3}{#2}{#1}\eq@paddingmax}
3890 \eq@define@key\eq@keycat{indent}[2em]{%
3891   \def\eq@indent@val{#1}}
3892 \eq@define@key\eq@keycat{indent*}[2em]{%
3893   \def\eq@indent@val{#1}\def\eq@paddingleft@val{#1}}

```

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

```

3894 \def\eq@keycat{equations, equationsbox, setup}
3895 \eq@define@key\eq@keycat{classout}{\eq@class@innerleft@set{#1}}
3896 \eq@define@key\eq@keycat{classin}{\eq@class@innerright@set{#1}}
3897 \eq@define@key\eq@keycat{classin*}{\eq@class@innerlead@set{#1}}
3898 \eq@define@key\eq@keycat{ampeq}[]{\eq@class@ampeq}
3899 \eq@define@key\eq@keycat{eqamp}[]{\eq@class@eqamp}
3900 \eq@define@key\eq@keycat{class}{\eq@decide@select{#3}{#2}{#1}{%
3901   {ampeq, amprel, eqafter, beforel}\eq@class@ampeq}, %
3902   {eqamp, relamp, eqbefore, afterrel}\eq@class@eqamp}}

```

Punctuation. **TODO:** describe

```

3903 \let\eq@punct@main\relax
3904 \def\eq@keycat{equations, equationsbox, setup}
3905 \eq@define@key\eq@keycat{punctsep}[\,]{\def\eq@punct@sep{#1}}
3906 \eq@define@key\eq@keycat{punct}[.]{\def\eq@punct@main{#1}}
3907 \eq@define@key\eq@keycat{punctline}[,]{\def\eq@punct@line{#1}}
3908 \eq@define@key\eq@keycat{punctcol}[,]{\def\eq@punct@col{#1}}
3909 \eq@define@key\eq@keycat{punct*}[]{\let\eq@punct@main\relax}
3910 \eq@define@key\eq@keycat{punctline*}[]{\let\eq@punct@line\relax}
3911 \eq@define@key\eq@keycat{punctcol*}[]{\let\eq@punct@col\relax}

```

Global Switches. **TODO:** describe

```

3912 \let\eq@multi@linesfallback\eq@true
3913 \let\eq@single@crerror\eq@true
3914 \let\eq@ampproof@active\eq@false
3915 \eq@define@key{setup}{linesfallback}[true]{%
3916   \eq@decide@bool{#3}{#2}{#1}\eq@multi@linesfallback}
3917 \eq@define@key{setup}{ampproof}[true]{%
3918   \eq@decide@bool{#3}{#2}{#1}\eq@ampproof@active}

```

```

3919 \eql@define@key{setup}{crrror}[true]{%
3920   \eql@decide@bool{#3}{#2}{#1}\eql@single@crrror}
3921 \eql@define@key{equations,setup}{rescan}[true]{%
3922   \eql@decide@if{#3}{#2}{#1}%
3923   {\let\eql@scan@body\eql@scan@body@rescan}%
3924   {\let\eql@scan@body\eql@scan@body@dump}}
3925 \eql@define@key{setup}{defaults}{%
3926   \eql@decide@select{#3}{#2}{#1}{%
3927     {\classic}{\eql@defaults@classic}},%
3928     {\eqnlines}{\eql@defaults@eqnlines}}}}

```

Package Options. **TODO:** describe

```

3929 \let\eql@provide@opt@equation\eql@true
3930 \let\eql@provide@opt@amsmathends\eql@true
3931 \let\eql@provide@opt@amsmath\eql@true
3932 \let\eql@provide@opt@ang\eql@true
3933 \let\eql@provide@opt@eqref\eql@true
3934 \eql@define@key{setup}{equation}[true]{%
3935   \eql@error@packageoption{#2}%
3936   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@equation}
3937 \eql@define@key{setup}{amsmathends}[true]{%
3938   \eql@error@packageoption{#2}%
3939   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
3940 \eql@define@key{setup}{amsmath}[true]{%
3941   \eql@error@packageoption{#2}%
3942   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmath}
3943 \eql@define@key{setup}{ang}[true]{%
3944   \eql@error@packageoption{#2}%
3945   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
3946 \eql@define@key{setup}{eqref}[true]{%
3947   \eql@error@packageoption{#2}%
3948   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Q.5 Component Selection

TODO: describe

```

3949 \newenvironment{eql@gathered}
3950   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
3951 \newenvironment{eql@multlined}
3952   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
3953 \newenvironment{eql@aligned}
3954   {\eqnaddopt{align}\equationsbox}{\endequationsbox}

```

TODO: describe

```

3955 \newenvironment{eql@equation}
3956   {\eqnaddopt{equation}\equations}{\endequations}
3957 \newenvironment{eql@gather}
3958   {\eqnaddopt{lines}\equations}{\endequations}
3959 \newenvironment{eql@multiline}
3960   {\eqnaddopt{lines,padmax,shape=steps,numberline=out}\equations}
3961   {\endequations}
3962 \newenvironment{eql@align}
3963   {\eqnaddopt{align}\equations}{\endequations}
3964 \newenvironment{eql@flalign}
3965   {\eqnaddopt{align,margins=false}\equations}{\endequations}

```

```

3966 \newenvironment{eql@equation*}
3967   {\eqnaddopt{nonumber}\eql@equation}{\endequations}
3968 \newenvironment{eql@gather*}
3969   {\eqnaddopt{nonumber}\eql@gather}{\endequations}
3970 \newenvironment{eql@multline*}
3971   {\eqnaddopt{nonumber}\eql@multline}{\endequations}
3972 \newenvironment{eql@align*}
3973   {\eqnaddopt{nonumber}\eql@align}{\endequations}
3974 \newenvironment{eql@flalign*}
3975   {\eqnaddopt{nonumber}\eql@flalign}{\endequations}

```

TODO: describe

```

3976 \def\eql@provide@movecmd#1#2{%
3977   \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
3978 }
3979 \def\eql@provide@undefinecmd#1{%
3980   \expandafter\let\csname #1\endcsname\@undefined
3981 }
3982 \def\eql@provide@moveenv#1#2{%
3983   \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
3984   \expandafter\let\csname end#1\expandafter\endcsname\csname end#2\endcsname
3985 }
3986 \def\eql@provide@undefineenv#1{%
3987   \expandafter\let\csname #1\endcsname\@undefined
3988   \expandafter\let\csname end#1\endcsname\@undefined
3989 }

```

TODO: describe

```

3990 \def\eql@provide@onlyonce#1#2{%
3991   \def\eql@tmp{#2}%
3992   \def\@tempa{#1}%
3993   \ifx\eql@tmp\@tempa
3994     \let\eql@tmp\@undefined
3995   \fi
3996   \ifx\eql@tmp\@empty
3997     \let\eql@tmp\@undefined
3998   \fi
3999   \def\@tempa{*}%
4000   \ifx\eql@tmp\@tempa
4001     \def\eql@tmp{#1}%
4002   \fi
4003   \ifdefined\eql@tmp\else
4004     \ifcsname eql@provided@#1\endcsname
4005       \def\eql@tmp{#1}%
4006     \else
4007       \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4008     \fi
4009   \fi
4010 }

```

TODO: describe

```

4011 \def\eql@provide@cmdonlyonce#1#2{%
4012   \def\eql@tmp{#2}%
4013   \edef\@tempb{\expandafter\noexpand\csname#1\endcsname}%
4014   \ifx\eql@tmp\@tempb
4015     \let\eql@tmp\@undefined
4016   \fi
4017   \ifx\eql@tmp\@empty

```

```

4018   \let\eql@tmp\@undefined
4019   \fi
4020   \def\@tempa{*}%
4021   \ifx\eql@tmp\@tempa
4022     \let\eql@tmp\@tempb
4023   \fi
4024   \ifdefined\eql@tmp
4025     \edef\eql@tmp{\expandafter\expandafter\expandafter\@gobble
4026       \expandafter\string\eql@tmp}%
4027   \else
4028     \ifcsname eql@provided@#1\endcsname
4029       \let\eql@tmp\@tempb
4030     \else
4031       \expandafter\let\csname eql@provided@#1\endcsname\eql@true
4032     \fi
4033   \fi
4034 }

```

TODO: describe

```

4035 \def\eql@provide@cmd#1#2{%
4036   \eql@provide@cmdonlyonce{#1}{#2}%
4037   \ifdefined\eql@tmp
4038     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@#1}%
4039   \else
4040     \eql@amsmath@after{%
4041       \eql@provide@movecmd{ams#1}{#1}%
4042       \eql@provide@movecmd{#1}{eql@#1}%
4043     }%
4044     \AddToHook{package/mathtools/after}{%
4045       \eql@provide@movecmd{#1}{eql@#1}%
4046     }%
4047     \eql@provide@movecmd{#1}{eql@#1}%
4048     \eql@amsmath@before{\eql@provide@undefinecmd{#1}}%
4049   \fi
4050 }

```

TODO: describe

```

4051 \def\eql@amsmath@endfix#1#2{%
4052   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
4053   \long\edef\@tempb{\expandafter\noexpand\csname eql@amsmath@end#2\endcsname}%
4054   \expandafter\ifx\csname end#1\endcsname\@tempa
4055     \expandafter\let\csname end#1\endcsname\@tempb
4056   \fi
4057 }

```

TODO: describe

```

4058 \def\eql@amsmath@fixends{%
4059   \eql@amsmath@after{%
4060     \let\eql@amsmath@endmultiline\endmultiline
4061     \eql@amsmath@endfix{multiline*}{multiline}%
4062     \let\eql@amsmath@endgather\endgather
4063     \eql@amsmath@endfix{gather*}{gather}%
4064     \let\eql@amsmath@endalign\endalign
4065     \eql@amsmath@endfix{align*}{align}%
4066     \eql@amsmath@endfix{flalign}{align}%
4067     \eql@amsmath@endfix{flalign*}{align}%
4068     \eql@amsmath@endfix{alignat}{align}%
4069     \eql@amsmath@endfix{alignat*}{align}%

```

```

4070 \eql@amsmath@endfix{xalignat}{align}%
4071 \eql@amsmath@endfix{xalignat*}{align}%
4072 \eql@amsmath@endfix{xxalignat}{align}%
4073 \let\eql@amsmath@endaligned\endaligned
4074 \eql@amsmath@endfix{gathered}{aligned}%
4075 \eql@amsmath@endfix{alignedat}{aligned}%
4076 }
4077 }

```

TODO: describe

```

4078 \def\eql@provide@env#1#2{%
4079 \eql@provide@onlyonce{#1}{#2}%
4080 \ifdefined\eql@tmp
4081 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@#1}%
4082 \else
4083 \eql@amsmath@after{%
4084 \eql@provide@moveenv{ams#1}{#1}%
4085 \eql@provide@moveenv{ams#1*}{#1*}%
4086 \eql@provide@moveenv{#1}{eql@#1}%
4087 \eql@provide@moveenv{#1*}{eql@#1*}%
4088 }%
4089 \AddToHook{package/mathtools/after}{%
4090 \eql@provide@moveenv{#1}{eql@#1}%
4091 \eql@provide@moveenv{#1*}{eql@#1*}%
4092 }%
4093 \eql@provide@moveenv{#1}{eql@#1}%
4094 \eql@provide@moveenv{#1*}{eql@#1*}%
4095 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
4096 \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
4097 \fi
4098 }

```

TODO: describe

```

4099 \def\eql@provide@env@equation#1{%
4100 \eql@provide@onlyonce{equation}{#1}%
4101 \ifdefined\eql@tmp
4102 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@equation}%
4103 \else
4104 \eql@amsmath@after{%
4105 \eql@provide@moveenv{amsequation}{equation}%
4106 \eql@provide@moveenv{amsequation*}{equation*}%
4107 \eql@provide@moveenv{equation}{eql@equation}%
4108 \eql@provide@moveenv{equation*}{eql@equation*}%
4109 }%
4110 \AddToHook{package/hyperref/after}{%
4111 \@ifpackageloaded{amsmath}{}{%
4112 \let\latexequation\H@equation
4113 \let\endlatexequation\H@endequation
4114 \eql@provide@moveenv{hyperrefequation}{equation}%
4115 \eql@provide@moveenv{equation}{eql@equation}%
4116 }%
4117 }%
4118 \@ifpackageloaded{amsmath}{}{\@ifpackageloaded{hyperref}{}{%
4119 \eql@provide@moveenv{latexequation}{equation}%
4120 }%
4121 \eql@provide@moveenv{equation}{eql@equation}%
4122 \eql@provide@moveenv{equation*}{eql@equation*}%
4123 \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%

```

```

4124 \ifdefined\eql@tagging@on
4125   \AddToHook{begindocument/end}{%
4126     \eql@provide@moveenv{equation}{\eql@equation}%
4127     \eql@provide@moveenv{equation*}{\eql@equation*}%
4128   }%
4129 \fi
4130 \fi
4131 }

```

TODO: describe

```

4132 \def\eql@provide@env@multlined#1{%
4133   \eql@provide@onlyonce{multlined}{#1}%
4134   \ifdefined\eql@tmp
4135     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@multlined}%
4136   \else
4137     \AddToHook{package/mathtools/after}{%
4138       \eql@provide@moveenv{amsmultlined}{\multlined}%
4139       \eql@provide@moveenv{multlined}{\eql@multlined}%
4140     }%
4141     \eql@provide@moveenv{multlined}{\eql@multlined}%
4142     \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
4143       \eql@provide@undefineenv{multlined}}}%
4144   \fi
4145 }

```

TODO: describe

```

4146 \def\eql@provide@env@subequations#1{%
4147   \eql@provide@onlyonce{subequations}{#1}%
4148   \ifdefined\eql@tmp
4149     \expandafter\eql@provide@moveenv
4150     \expandafter{\eql@tmp}{\eql@subequations}%
4151   \else
4152     \eql@amsmath@after{%
4153       \eql@provide@moveenv{amssubequations}{\subequations}%
4154       \eql@provide@moveenv{subequations}{\eql@subequations}%
4155     }%
4156     \AddToHook{package/hyperref/after}{%
4157       \AddToHook{cmd/subequations/before}[hyperref]{}%
4158       \AddToHook{cmd/subequations/after}[hyperref]{}%
4159       \RemoveFromHook{cmd/subequations/before}[hyperref]%
4160       \RemoveFromHook{cmd/subequations/after}[hyperref]%
4161       \AddToHook{cmd/amssubequations/before}%
4162       {%
4163         \stepcounter{equation}%
4164         \protected@edef\theHparentequation{\theHequation}%
4165         \addtocounter{equation}{-1}%
4166       }%
4167       \AddToHook{cmd/amssubequations/after}%
4168       {%
4169         \def\theHequation{\theHparentequation\alph{equation}}%
4170         \ignorespaces
4171       }%
4172       \AddToHook{begindocument/end}{%
4173         \eql@provide@moveenv{subequations}{\eql@subequations}%
4174       }%
4175     }%
4176     \eql@provide@moveenv{subequations}{\eql@subequations}%
4177     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%

```

```
4178 \fi
4179 }
```

TODO: describe

```
4180 \def\eq@provide@sqr{%
4181 \let\[\eq@equations@sqr@open
4182 \let\]\eq@equations@sqr@close
4183 \eq@amsmath@after{%
4184 \let\[\eq@equations@sqr@open
4185 \let\]\eq@equations@sqr@close
4186 }%
4187 \ifdefined\eq@tagging@on
4188 \AddToHook{begindocument/end}{%
4189 \let\[\eq@equations@sqr@open
4190 \let\]\eq@equations@sqr@close
4191 }%
4192 \fi
4193 }
```

TODO: describe

```
4194 \def\eq@provide@ang{%
4195 \let\<\eq@equations@ang@open
4196 \let\>\eq@equations@ang@close
4197 }
```

TODO: describe

```
4198 \eq@define@key{provide}{equation} [] {\eq@provide@env{equation}{#1}}
4199 \eq@define@key{provide}{gather} [] {\eq@provide@env{gather}{#1}}
4200 \eq@define@key{provide}{multline} [] {\eq@provide@env{multline}{#1}}
4201 \eq@define@key{provide}{align} [] {\eq@provide@env{align}{#1}}
4202 \eq@define@key{provide}{flalign} [] {\eq@provide@env{flalign}{#1}}
4203 \eq@define@key{provide}{aligned} [] {\eq@provide@env{aligned}{#1}}
4204 \eq@define@key{provide}{gathered} [] {\eq@provide@env{gathered}{#1}}
4205 \eq@define@key{provide}{multlined} [] {\eq@provide@env{multlined}{#1}}
4206 \eq@define@key{provide}{subequations} [] {\eq@provide@env{subequations}{#1}}
4207 \eq@define@key{provide}{sqr} [] {\eq@provide@sqr}
4208 \eq@define@key{provide}{ang} [] {\eq@provide@ang}
4209 \eq@define@key{provide}{eqref} [] {\eq@provide@cmd{eqref}{#1}}
4210 \eq@define@key{provide}{notag} [] {\eq@provide@cmd{notag}{#1}}
4211 \eq@define@key{provide}{thetag} [] {\eq@provide@cmd{thetag}{#1}}
4212 \eq@define@key{provide}{allowdisplaybreaks} [] {%
4213 \eq@provide@cmd{allowdisplaybreaks}{#1}}
4214 \eq@define@key{provide}{numberwithin} [] {\eq@provide@cmd{numberwithin}{#1}}
4215 \eq@define@key{provide}{tagform} [] {%
4216 \def\tagform@##1{\maketag@@@{\eq@tag@form{#1}}}}
4217 \eq@define@key{provide}{maketag} [] {%
4218 \def\maketag@@@##1{\eq@tag@box{##1}}}
```

TODO: describe

```
4219 \newcommand{\eqnlinesprovide}[1]{%
4220 (dev)\eq@dev@start\eqnlinesprovide
4221 \eq@setkeys{provide}{#1}}
```

Q.6 Global and Package Options

Handle global and package options:

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

```
4222 \newcommand{\eqnlineset}[1]{%
4223 (dev)\eql@dev@start\eqnlineset
4224 \eql@setkeys{setup}{#1}}
```

Disable error message for exclusive package options:

```
4225 \let\eql@error@packageoption@gobble
```

Pass undeclared options on to keyval processing:

```
4226 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Process package options:

```
4227 \ProcessOptions
```

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

```
4228 \def\eql@error@packageoption#1{%
4229 \eql@error{may only use ‘#1’ as a package option}%
4230 }
```

Provide classes of interfaces:

```
4231 \ifdefined\tagsleft@true\else
4232 \expandafter\newif\csname iftagsleft@endcsname
4233 \fi
4234 \ifdefined@\fleqntrue\else
4235 \expandafter\newif\csname if@fleqn@endcsname
4236 \fi
4237 \ifdefined\eql@provide@opt@amsmath
4238 \let\eql@provide@opt@equation\eql@true
4239 \eql@amsmath@after{%
4240 \iftagsleft@
4241 \eqnlineset{tagsleft}
4242 \else
4243 \eqnlineset{tagsright}
4244 \fi
4245 \if@fleqn
4246 \eqnlineset{left}
4247 \else
4248 \eqnlineset{center}
4249 \fi
4250 }
4251 \fi
```

TODO: describe

```
4252 \ifdefined\eql@provide@opt@equation\eqnlinesprovide{equation,sqr}\fi
```

TODO: describe

```
4253 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

TODO: describe

```
4254 \ifdefined\eql@provide@opt@amsmath
4255 \eqnlinesprovide{%
4256 multiline,gather,align,flalign,%
4257 multlined,gathered,aligned,%
```

```
4258     subequations}
4259 \fi
```

TODO: describe

```
4260 \ifdefined\eqL@provide@opt@ang\eqnlinesprovide{ang}\fi
```

TODO: describe

```
4261 \ifdefined\eqL@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```