

# 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`

2025/02/25, v0.5

– 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  $\text{\LaTeX}$  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

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>4</b>
2.1	Equations Environment . . . . .	5
2.2	Numbering . . . . .	5
2.3	Horizontal Adjustment . . . . .	7
2.4	Punctuation . . . . .	8
2.5	Math Classes at Alignment . . . . .	8
2.6	Vertical Spacing . . . . .	9
2.7	Further Environments . . . . .	10
2.8	Feature Selection and Package Options . . . . .	11
<b>3</b>	<b>Information</b>	<b>11</b>
3.1	Copyright . . . . .	11
3.2	Credits . . . . .	12
3.3	Files and Installation . . . . .	12
3.4	Related CTAN Packages . . . . .	13
3.5	Feature Suggestions . . . . .	13
3.6	Revision History . . . . .	14
<b>A</b>	<b>Implementation</b>	<b>14</b>

<b>B</b>	<b>General Support</b>	<b>14</b>
B.1	Development Messages . . . . .	14
B.2	Look-Ahead in Alignment . . . . .	14
B.3	Error Messages . . . . .	16
B.4	amsmath Integration . . . . .	17
<b>C</b>	<b>Parameters and Registers</b>	<b>17</b>
C.1	Supporting Definitions . . . . .	17
C.2	Parameters . . . . .	17
C.3	Registers . . . . .	20
C.4	Hooks . . . . .	21
<b>D</b>	<b>Punctuation</b>	<b>22</b>
<b>E</b>	<b>Math Classes at Alignment</b>	<b>23</b>
<b>F</b>	<b>Equations Box Environment</b>	<b>24</b>
<b>G</b>	<b>Subequation Numbering</b>	<b>28</b>
<b>H</b>	<b>Equation Numbering</b>	<b>30</b>
H.1	Macros Shared with Amsmath . . . . .	30
H.2	Showkeys Integration . . . . .	31
H.3	Labels . . . . .	32
H.4	Tags . . . . .	32
H.5	Anchors . . . . .	33
H.6	Tag Composition . . . . .	34
H.7	Tagbox Methods . . . . .	35
H.8	Numbering Schemes . . . . .	37
H.9	Numbering Framework . . . . .	39
H.10	Subequations . . . . .	43
<b>I</b>	<b>Display Equations Support</b>	<b>44</b>
I.1	Display Breaks . . . . .	44
I.2	General Initialisation . . . . .	45
I.3	halign Support . . . . .	47
I.4	Stack . . . . .	48
<b>J</b>	<b>Horizontal Spacing for Lines</b>	<b>50</b>
J.1	Supporting Definitionss . . . . .	50
J.2	Shape Schemes . . . . .	50
J.3	Adjustment Methods . . . . .	53
J.4	Centered Layout . . . . .	54
J.5	Flush-Left Layout . . . . .	59
<b>K</b>	<b>Single-Line Equation</b>	<b>62</b>
K.1	Environment . . . . .	62
K.2	Native . . . . .	63
K.3	Adjustment . . . . .	63
<b>L</b>	<b>Multi-Line Support</b>	<b>64</b>
L.1	Registers . . . . .	64
L.2	Measure Support . . . . .	65
L.3	Print Support . . . . .	66
L.4	Line Breaks . . . . .	67

L.5	Intertext . . . . .	67
L.6	Main . . . . .	69
<b>M</b>	<b>Multi-Line Lines Mode</b>	<b>70</b>
M.1	Measure . . . . .	70
M.2	Print . . . . .	71
<b>N</b>	<b>Multi-Line Align Mode</b>	<b>73</b>
N.1	Registers . . . . .	73
N.2	General Processing, Preamble . . . . .	74
N.3	Print . . . . .	75
N.4	Measure . . . . .	77
N.5	Adjust . . . . .	79
<b>O</b>	<b>Interface</b>	<b>84</b>
O.1	Scanning the Environment's Body . . . . .	84
O.2	Options Processing . . . . .	87
O.3	Environments . . . . .	87
O.4	Independent Routines of <code>amsmath</code> . . . . .	89
<b>P</b>	<b>Options</b>	<b>90</b>
P.1	Selection Tools . . . . .	90
P.2	Declaration Code . . . . .	91
P.3	Options Declarations . . . . .	92
P.4	Component Selection . . . . .	97
P.5	Global and Package Options . . . . .	103

## 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 `\<...\>` 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 L<sup>A</sup>T<sub>E</sub>X macros) which will be explained further below.

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*)

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

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

`\[...]` The package offers several alternative names for the same mode as well as a symbolic short form `\<...>` extending the L<sup>A</sup>T<sub>E</sub>X display equation form `\[...]` to multi-line equations.

`~` (*key*) Here, the tilde ‘`~`’ in `\<~...>` is a modifier character which acts as a short form for the optional argument `lines` selecting the lines mode. Both short forms can be customised by 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`, `multline`, `gather`, `align` and `flalign` including their starred variants but neither the `alignat` alternative forms nor `split`. It is possible to define further equation environments *env* with a predefined set of options *opts* using:

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

## 2.2 Numbering

`numberline` (*key*) The package extends the established interface of L<sup>A</sup>T<sub>E</sub>X and the amsmath package for la-

bellings 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<sup>A</sup>T<sub>E</sub>X labels as usual and they can be turned into manually 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 optional arguments):  
`@ (key)`

`label={label},      tag[*]={tag},      \@{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 methods defined by `amsmath`, one which defines the layout and another one which adds a decoration (brackets).  
`tagform` (*key*) 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  
`mincolsep` (*key*) of subsequent columns forms a unit which is aligned at the intermediate alignment marker  
`maxcolsep` (*key*) ‘&’. 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` ( $\equiv\backslashqquad$ ) 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`

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

`padding` (*key*) Note that (hanging) indentation requires to add some padding around the equations block via  
`padleft` (*key*) the optional argument `paddingpadleftpadright[={dimen}]` or `padmax` to extend padding  
`padright` (*key*)  
`padmax` (*key*)

to the whole line. Note that `indent*={dimen}` sets the default indentation amount and the 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 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 \> 1
\<[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 some potential gaps of the surrounding text. The apparent height and depth are defined by `displayheight` (*key*) and `displaydepth` (*key*) 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 `TEX` feature for all single- and multi-line equation environments. `TEX` also reduces

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

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  
also take the values `above`, `below` or `both`.

`...skip` (*key*) Variable amounts of skip can be set via `aboveskip` and `belowskip` or `skip` for both simul-  
`\vspace` taneously. In addition, the package extends the `\vspace` mechanism of L<sup>A</sup>T<sub>E</sub>X to equation  
`...space` (*key*) bodies where it adds vertical space below the next equation line or below the equation en-  
vironment. Additional glue can be added above or below equation environments by means  
of the options `abovespace` and `belowspace`.

`...skip` (*key*) The package also maintains several global vertical space settings:

above	below	both	description
<code>aboveskip</code>	<code>belowskip</code>	<code>skip</code>	standard amount of glue
<code>aboveshortskip</code>	<code>belowshortskip</code>	–	reduced glue for short equations
<code>abovemedskip</code>	<code>belowmedskip</code>	<code>medskip</code>	medium amount of glue
<code>abovetagskip</code>	<code>belowtagskip</code>	<code>tagskip</code>	glue for outer raised/lowered tags
<code>abovetopskip</code>	–	–	glue at top of vertical list
<code>aboveparskip</code>	–	–	glue when starting a paragraph
<code>abovecontskip</code>	–	–	glue when started from empty paragraph

`spread` (*key*) Likewise, the spacing between the lines of a multi-line equation environment can be adjusted  
`strut` (*key*) via `spread={dimen}` which defaults to `\jot≐3pt`. In addition, all equation lines and tags  
`strutttag` (*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 envi-  
ronment. 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  
`margin` (*key*) arbitrary math content. It works analogously to `equations` including optional arguments  
`marginleft` (*key*) and modifiers, but it offers a reduced range of functionality such as (evidently) no number-  
`marginright` (*key*) ing. Additional arguments are given by `margin`, `marginleft`, `marginright` which specify  
additional margin space around the equations box.

`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 Feature Selection and Package Options

`\eqnlineset` Features and 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 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>\langle... \rangle</code>

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

```
\eqnlinesprovide{features}
```

*features* is a comma-separated list of features:

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

## 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X package `amsmath` (initially named `amstex`) which in turn is based on the T<sub>E</sub>X macro system `amstex` written by Michael Spivak. The initial work of porting `amstex` to L<sup>A</sup>T<sub>E</sub>X 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
- pdf tagging<sup>1</sup>

---

<sup>1</sup>See <https://latex3.github.io/tagging-project/>

## 3.6 Revision History

**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 \@currenenv}}
6 <dev>\def\eql@dev@leaveenv{\eql@dev{ leaving \@currenenv}}
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\eq@ifnextchar@tight#1#2#3{%
10 \let\reserved@d= #1%
11 \def\reserved@a{#2}%
12 \def\reserved@b{#3}%
13 \futurelet\@let@token\eq@ifnch@tight
14 }
15 \def\eq@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=0%
24 \makeatletter
25 \global\let\eq@atxii\tmp
26 \endgroup

```

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

```

27 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
28 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
29 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
30 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
31 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{0}{#1}{%
32 \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
33 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{0}{#1}{%
34 \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}}
35 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[#{1}-#{1}[#{2}]]}%]
36 \long\def\eq@testopt@tight#1#2{\eq@ifnextchar@tight[#{1}-#{1}[#{2}]]}%]
37 \long\def\eq@teststaropt@loose#1#2#3{%
38 \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}}
39 \long\def\eq@teststaropt@tight#1#2#3{%
40 \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}}

```

`\eq@spbgroup` The second challenge is addressed by enclosing the look-ahead in spurious groups<sup>2</sup> which `\eq@spegroup` protect against triggering ‘&’. The macros `\eq@spbgroup` and `\eq@spegroup` open and `\eq@srbgroup` close a spurious group. For some reason, the look-ahead mechanism requires further `\eq@sregroup` protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end. These adjustments are included in the macros `\eq@srbgroup` and `\ers@spegroup`:

```

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

```

<sup>2</sup>See <https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3040>, <https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=amslatex/1834> and <https://tex.stackexchange.com/questions/9897/showcase-of-brace-tricks-egroup-iffalse-fi-etc>.

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

```
45 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1{\eql@sregroup#2}}
46 \long\def\eql@ampprotecttwo#1#2#3{%
47   \eql@srbgroup#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}}
52 \def\eql@ifstar@loose@ampsafe{\eql@ampprotecttwo\eql@ifstar@loose}
53 \def\eql@ifstar@tight@ampsafe{\eql@ampprotecttwo\eql@ifstar@tight}
54 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
55 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
56 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
57 \long\def\eql@teststaropt@tight@ampsafe{%
58   \eql@ampprotecttwo\eql@teststaropt@tight}}
```

`\eql@amproof` We may want to replace L<sup>A</sup>T<sub>E</sub>X’s definitions `\@ifnextchar`, `\@ifstar` and `\@testopt` to `\eql@amprevert` respect ‘&’ characters within aligned equations. This might make unrelated definitions with optional arguments and starred variants more robust in this context. The macro `\eql@amproof` overwrites the original definitions, and `\eql@amprevert` reverts the changes:

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

### B.3 Error Messages

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

```
\eql@warning
72 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
73 \def\eql@warning{\PackageWarning{eqnlines}}
```

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

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

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

```
\eql@warn@label@multiple
76 \def\eql@warn@label@unused{\eql@warning{Unused equation \string\label:
77   label ‘\eql@nextlabel’ will be lost}}
78 \def\eql@warn@label@multiple#1{\eql@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@futurebefore` will execute argument #1 before any future loading of `amsmath`: **TODO**: might use `\@for` for undefine

```

84 \def\eq@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
85 \def\eq@amsmath@futurebefore#1{%
86   \ifpackageloaded{amsmath}{}\{\AddToHook{package/amsmath/before}{#1}}\}

```

## C Parameters and Registers

In the following, we collect parameter and register definitions.

### C.1 Supporting Definitions

`\eq@false` (*bool*) Rather than the standard L<sup>A</sup>T<sub>E</sub>X 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  $\epsilon$ -T<sub>E</sub>X conditional `\ifdefined\xxx`. In order to make the scheme more tangible, we define the two expected values for boolean variables:

```

87 \let\eq@false\@undefined
88 \let\eq@true\@empty

```

`\eq@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 `\eq@regsetdelayed` performs the delayed initialisation:

```

89 \newskip\eq@regunset@
90 \eq@regunset@-1sp\relax
91 \def\eq@regsetdelayed#1#2{%
92   #1\eq@regunset@\relax
93   \AtBeginDocument{%
94     \ifdim#1=\eq@regunset@
95       \setlength#1{#2}%
96     \fi
97   }%
98 }

```

### 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:

```
99 \let\eql@tagsleft\eql>false
```

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

```
100 \let\eql@flushleft\eql>false
```

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

```
101 \newdimen\eql@flushleftmargin@
102 \newdimen\eql@flushleftmarginmin@
103 \newdimen\eql@flushleftmarginmax@
104 \eql@regsetdelayed\eql@flushleftmargin@\leftmargini
105 \eql@flushleftmarginmin@\z@
106 \eql@flushleftmarginmax@.5\maxdimen
```

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

```
107 \newdimen\eql@tagmargin@
108 \eql@tagmargin@\z@
```

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

```
109 \newdimen\eql@indent@
110 \def\eql@indent@val{2em}
```

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

```
111 \newdimen\eql@paddingleft@
112 \newdimen\eql@paddingright@
113 \def\eql@paddingleft@val{0pt}
114 \def\eql@paddingright@val{0pt}
```

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

```
115 \let\eql@paddingmax\eql>false
```

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

```
116 \def\eql@box@marginleft{\z@skip}
117 \def\eql@box@marginright{\z@skip}
```

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

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

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

```
119 \def\eql@spread{\jot}
```

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

```
120 \newdimen\eql@tagfuzz@
121 \eql@tagfuzz@16sp\relax
```

`\abovedisplayskip@` (*skip*) We need two registers `\eql@abovedisplayskip@` and `\eql@belowdisplayskip@` to feed values to the final adjustment section after the end of `\halign` where `\setlength` of `calc` would fail:

```
122 \newskip\eql@abovedisplayskip@
123 \newskip\eql@belowdisplayskip@
```

`\abovedisplaytagskip@` (*skip*)

`\belowdisplaytagskip@` (*skip*)

```
124 \newskip\eql@abovedisplaytagskip@
125 \newskip\eql@belowdisplaytagskip@
126 \eql@abovedisplaytagskip@\z@skip
127 \eql@belowdisplaytagskip@\z@skip
```

`\displayshortmode@` (*counter*) The setting `\eql@displayshortmode@` 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:

```
128 \newcount\eql@displayshortmode@
129 \eql@displayshortmode@\tw@
```

`\displaycontskip@` (*skip*) The glue when an equation is at the top of a horizontal list is specified by `\eql@abovedisplaycontskip@`:

```
130 \newskip\eql@abovedisplaycontskip@
131 \eql@regsetdelayed\eql@abovedisplaycontskip@\abovedisplayshortskip
```

`\abovedisplaytopskip@` (*skip*) The glue when an equation is at the top of a vertical list is specified by

`\belowdisplaytopskip@` (*skip*) `\eql@abovedisplaytopskip@` and `\eql@belowdisplaytopskip@`:

```
132 \newskip\eql@abovedisplaytopskip@
133 \newskip\eql@belowdisplaytopskip@
134 \eql@abovedisplaytopskip@\z@skip
135 \eql@regsetdelayed\eql@belowdisplaytopskip@\belowdisplayskip
```

`\displayparskip@` (*skip*) The glue when an equation starts a paragraph is specified by

`\eql@abovedisplaytopskip@`:

```
136 \newskip\eql@abovedisplayparskip@
137 \eql@regsetdelayed\eql@abovedisplayparskip@\abovedisplayskip
```

`\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:

```

138 \def\eqldisplay@height{\ht\eql@strutbox@}
139 \def\eqldisplay@depth{\dp\eql@strutbox@}

```

`\displaymedskip@` (*skip*) The surrounding glue for an equation with reduced spacing is given by `\eql@abovedisplaymedskip@` and `\eql@belowdisplaymedskip@`:

```

140 \newskip\eql@abovedisplaymedskip@
141 \newskip\eql@belowdisplaymedskip@
142 \eql@regsetdelayed\eql@abovedisplaymedskip@\belowdisplayshortskip
143 \eql@regsetdelayed\eql@belowdisplaymedskip@\eql@belowdisplaymedskip@

```

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

```

144 \newdimen\eql@colsepmin@
145 \newdimen\eql@colsepmax@
146 \def\eql@colsepmin@val{1em}
147 \def\eql@colsepmax@val{.5\maxdimen}

```

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

```

148 \newdimen\eql@tagwidthmin@
149 \eql@tagwidthmin@\z@

```

`\eql@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eql@tagsepmin@`. T<sub>E</sub>X'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

```

150 \newdimen\eql@tagsepmin@
151 \eql@regsetdelayed\eql@tagsepmin@{.5em}

```

### C.3 Registers

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

```

152 \newcount\eql@row@
153 \newcount\eql@totalrows@

```

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

```

154 \newdimen\eql@colsep@

```

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

```

155 \newdimen\eqldisplay@firstavail@
156 \def\eqldisplay@firstavail@set#1{%
157   \ifnum\eql@row@=\@ne
158     \global\eqldisplay@firstavail@#1%

```

```

159 \fi
160 }

```

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

```

161 \newcount\eql@raisetag@firstlast@

```

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

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

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

```

162 \newbox\eql@fieldbox@
163 \newbox\eql@tagbox@
164 \newdimen\eql@fieldwidth@
165 \newdimen\eql@tagwidth@

```

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

```

166 \newdimen\eql@line@height@
167 \newdimen\eql@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:

```

168 \newif\ifmeasuring@
169 \AddToHook{package/hyperref/after}{
170   \ifdefined\Hy@ifnotmeasuring
171     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@else#1\fi}
172   \fi
173 }

```

`\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:

```

174 \ifdefined\@displaytrue\else
175 \everydisplay\expandafter{\the\everydisplay\@displaytrue}
176 \fi
177 \newif\if@display

```

## C.4 Hooks

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

```

178 \let\eql@hook@blockbefore\@empty
179 \let\eql@hook@blockafter\@empty
180 \let\eql@hook@blockin\@empty
181 \let\eql@hook@blockout\@empty

```

```

182 \let\eqL@hook@linein\@empty
183 \let\eqL@hook@lineout\@empty
184 \let\eqL@hook@colin\@empty
185 \let\eqL@hook@colout\@empty
186 \let\eqL@hook@eqin\@empty
187 \let\eqL@hook@eqout\@empty
188 \let\eqL@hook@innerleft\@empty
189 \let\eqL@hook@innerright\@empty
190 \let\eqL@hook@innerlead\@empty

```

## D Punctuation

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

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

```

\eqL@punct@line
\eqL@punct@block
191 \let\eqL@punct@col\@empty
192 \let\eqL@punct@line\relax
193 \let\eqL@punct@block\relax

```

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

```

194 \let\eqL@punct@sep\relax

```

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

```

\eqnpunct
195 \def\eqnpunctcol{\eqL@ifstar@tight\eqL@punct@col@setrelax\eqL@punct@col@set}
196 \def\eqL@punct@col@set#1{\def\eqL@punct@col{#1}\ignorespaces}
197 \def\eqL@punct@col@setrelax{\let\eqL@punct@col\@empty\ignorespaces}
198 \def\eqnpunctline{\eqL@ifstar@tight\eqL@punct@line@setrelax\eqL@punct@line@set}
199 \def\eqL@punct@line@set#1{\def\eqL@punct@line{#1}\ignorespaces}
200 \def\eqL@punct@line@setrelax{\let\eqL@punct@line\relax\ignorespaces}
201 \def\eqnpunctmain{\eqL@ifstar@tight\eqL@punct@main@setrelax\eqL@punct@main@set}
202 \def\eqL@punct@main@set#1{\eqnlineset{punct=#1}\ignorespaces}
203 \def\eqL@punct@main@setrelax{\eqnlineset{punct*}\ignorespaces}
204 \def\eqnpunct{\eqL@ifstar@tight\eqL@punct@next@setrelax\eqL@punct@next@set}
205 \def\eqL@punct@next@set#1{\eqnaddopt{punct=#1}\ignorespaces}
206 \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:

```

207 \def\eqL@punct@apply@col{%
208   \ifx\eqL@punct@col\@empty\else
209     \eqL@punct@sep
210     \eqL@punct@col
211     \let\eqL@punct@col\@empty
212   \fi
213 }

```

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

`eql@punct@apply@line`

```
214 \def\eql@punct@apply@line{%
215   \ifx\eql@punct@line\relax
216   % \TODO hand down immediately?
217   \else
218     \ifx\eql@punct@line\@empty\else
219       \eql@punct@sep
220       \eql@punct@line
221     \fi
222     \let\eql@punct@line\relax
223     \let\eql@punct@col\@empty
224   \fi
225 }
```

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

```
226 \def\eql@punct@apply@block{%
227   \ifx\eql@punct@block\relax
228   % \TODO hand down immediately?
229   \else
230     \ifx\eql@punct@block\@empty\else
231       \eql@punct@sep
232       \eql@punct@block
233     \fi
234     \let\eql@punct@block\relax
235     \let\eql@punct@line\relax
236     \let\eql@punct@col\@empty
237   \fi
238 }
```

## E Math Classes at Alignment

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

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

```
239 \def\eql@class@innerright@sel@{%
240   \ifdim\eql@fieldwidth@=\z@
241     \eql@class@innerlead
242   \else
243     \eql@class@innerright
244   \fi
245 }
```

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

```
246 \def\eql@class@innerleft@set#1{%
247   \def\eql@class@innerleft{#1}%
248 }
249 \def\eql@class@innerright@set#1{%
250   \def\eql@class@innerright{#1}%
251   \let\eql@class@innerright@sel\eql@class@innerright
252 }
```

```

253 \def\eq@class@innerlead@set#1{%
254   \def\eq@class@innerlead{#1}%
255   \let\eq@class@innerright@sel\eq@class@innerright@sel@
256 }

```

`\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`):

```

257 \def\eq@class@ampeq{%
258   \eq@class@innerleft@set{}%
259   \eq@class@innerright@set{{}}%
260 }
261 \def\eq@class@eqamp{%
262   \eq@class@innerleft@set{\mathrel{}}%
263   \eq@class@innerright@set{\mathrel{}}%
264   \eq@class@innerlead@set{{}}%
265 }
266 \eq@class@ampeq

```

## F Equations Box Environment

**TODO:** describe

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

`\eq@box@cr`

```

267 \protected\def\eq@box@cr{%
268   \eq@ampprotect\eq@testopt@tight\eq@box@cr@z@
269 }
270 \def\eq@box@cr@[#1]{%
271   \eq@punct@apply@line
272   \eq@hook@lineout
273   \eq@box@lastfield
274   \cr
275   \noalign{%
276     \setlength\skip@{#1}%
277     \vskip\skip@
278   }%
279 }

280 \let\eq@box@box\vcenter

281 \def\eq@box@lastfield@odd{%
282   &\omit
283   \kern-\wd\eq@fieldbox@
284   \box\eq@fieldbox@
285   \hfil
286   &\omit\kern-\eq@colsep@
287 }%
288 \def\eq@box@lastfield@even{&\omit\kern-\eq@colsep@}
289 \def\eq@box@lastfield@lines{&\omit\kern-2\eq@colsep@}

290 \def\eq@box@open@align{%
291   % \TODO templates
292   \let\eq@box@lastfield\@empty
293   \eq@halign@init{%
294     (dev)\eq@dev{starting new line}%

```



```

295 }%
296 \tabskip\z@skip
297 \halign\bgroup
298   &%
299   \let\eq@box@lastfield\eq@box@lastfield@odd
300   \global\setbox\eq@fieldbox@\hbox{%
301     \eq@strut@field
302     \@lign
303     $\m@th\displaystyle
304     \eq@hook@colin
305     ##%
306     \eq@class@innerleft
307     \eq@hook@innerleft
308     $%
309   }%
310   \hfil
311   \kern\wd\eq@fieldbox@
312   \tabskip\z@skip
313   &%
314   \eq@fieldwidth@\wd\eq@fieldbox@
315   \kern-\eq@fieldwidth@
316   \box\eq@fieldbox@
317   \let\eq@box@lastfield\eq@box@lastfield@even
318   \llap{\unhbox\eq@fieldbox@}%
319   \hbox{%
320     \eq@strut@field
321     \@lign
322     $\m@th\displaystyle
323     \eq@hook@innerright
324     \eq@class@innerright@sel
325     ##%
326     \eq@punct@apply@col
327     \eq@hook@colout
328     $%
329   }%
330   \hfil
331   \tabskip\eq@colsep@
332   \crrc
333   \noalign{%
334     \eq@hook@blockbefore
335   }%
336   \eq@hook@blockin
337 }

338 \def\eq@box@open@lines{%
339 % \TODO templates
340 \let\shoveleft\eq@adjust@shoveleft
341 \let\shovecenter\eq@adjust@shovecenter
342 \let\shoveright\eq@adjust@shoveright
343 \let\eq@box@lastfield\eq@box@lastfield@lines
344 \eq@halign@init{%
345 (dev)\eq@dev{starting line \the\eq@row@}%
346 \global\advance\eq@row@\@ne
347 }%
348 \tabskip\z@skip
349 \halign\bgroup
350   &%
351   \eq@shape@pos@\m@ne
352   \setbox\eq@fieldbox@\hbox{%

```

```

353 \eql@strut@field
354 \@lign
355 $\m@th\displaystyle
356 \eql@hook@colin
357 ##%
358 \eql@punct@apply@col
359 \eql@hook@colout
360 $%
361 }%
362 \ifnum\eql@shape@pos@=\m@ne
363 \eql@shape@eval
364 \fi
365 \ifcase\eql@shape@pos@
366 \kern\eql@shape@amount@
367 \box\eql@fieldbox@
368 \skip@\@flushglue
369 \advance\skip@\eql@paddingleft@\relax
370 \advance\skip@\eql@paddingright@\relax
371 \advance\skip@-\eql@shape@amount@\relax
372 \hskip\skip@
373 \or
374 \skip@\@flushglue
375 \advance\skip@\eql@paddingleft@\relax
376 \hskip\skip@
377 \box\eql@fieldbox@
378 \skip@\@flushglue
379 \advance\skip@\eql@paddingright@\relax
380 \hskip\skip@
381 \or
382 \skip@\@flushglue
383 \advance\skip@\eql@paddingleft@\relax
384 \advance\skip@\eql@paddingright@\relax
385 \hskip\skip@
386 \box\eql@fieldbox@
387 \fi
388 \tabskip\eql@colsep@
389 \crrr
390 \noalign{%
391 \eql@hook@blockbefore
392 }%
393 \eql@hook@blockin
394 }

395 \def\eql@box@close{%
396 \ifvmode\else
397 \global\eql@totalrows@\eql@row@
398 \eql@punct@apply@block
399 \eql@box@cr@[\z@skip]%
400 \fi
401 \crrr
402 \noalign{%
403 \eql@hook@blockafter
404 }%
405 \egroup
406 }

```

`\eql@box@start`

```

407 \def\eql@box@start{%

```

```

408 \relax
409 \ifmode
410   \let\eql@box@endmath\@empty
411 \else
412   $\let\eql@box@endmath=$%
413 \fi
414 \eql@nextopt@process{equationsbox}%
415 \let\eql@punct@block\eql@punct@main
416 \let\eql@punct@main\relax
417 \setlength\eql@colsep@eql@box@colsep
418 \setlength\eql@paddingleft@eql@paddingleft@val
419 \setlength\eql@paddingright@eql@paddingright@val
420 \setlength\eql@indent@eql@indent@val
421 \eql@stack@save@boxed
422 \let\eql@flushleft\eql@false
423 \eql@row@z@
424 \eql@totalrows@\@M
425 \eql@shape@sel
426 \setlength\skip@\eql@box@marginleft
427 \hskip\skip@
428 \eql@box@box@bgroup
429 \eql@display@leave
430 \let\\\eql@box@cr
431 \ifdefined\eql@box@mode@lines
432   \expandafter\eql@box@open@lines
433 \else
434   \expandafter\eql@box@open@align
435 \fi
436 }

```

\eql@box@end

```

437 \newcommand{\eql@box@end}{%
438   \eql@box@close
439   \egroup
440   \setlength\skip@\eql@box@marginright
441   \hskip\skip@
442   \eql@stack@restore
443   \eql@box@endmath
444 }

```

equationsbox (*env.*)

```

445 \newenvironment{equationsbox}{%
446 (dev)\eql@dev@enterenv
447 \eql@ampprotect\eql@box@testall\eql@box@start
448 }{%
449 \eql@box@end
450 (dev)\eql@dev@leaveenv
451 }

452 \def\eql@box@testall{\eql@box@testtilde}
453 \def\eql@box@testtilde#1{%
454   \eql@ifnextgobble@tight~%
455   {\eqnadopt{lines}\eql@box@testopt{#1}}%
456   {\eql@box@testopt{#1}}}
457 \def\eql@box@testopt#1{%
458   \eql@ifnextchar@tight [%]
459   {\eql@box@adopt{#1}}%

```

```

460     {#1}}
461 \def\eql@box@adopt#1[#2]{\eqnadopt{#2}#1}

462 \def\eql@mode@aligned{\let\eql@box@mode@lines\eql@false}
463 \def\eql@mode@lined{\let\eql@box@mode@lines\eql@true}
464 \eql@mode@aligned

```

## G Subequation Numbering

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

`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:

```

465 \eql@amsmath@futurebefore{
466   \let\c@parentequation\undefined
467   \let\theparentequation\undefined
468 }
469 \ifdefined\c@parentequation\else
470 \newcounter{parentequation}
471 \fi

```

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

```

472 \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:

```

473 \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`:

```

474 \def\eql@subequations@init{%
475   \edef\eql@subequations@restorecounter{%
476     \global\c@equation\the\c@equation\relax}%
477   \ifdefined\eql@blocktag
478     \expandafter\eql@tag@makenext\eql@blocktag
479     \eql@nexttag
480     \global\advance\eql@numbering@refcount@\@ne
481     \protected@edef\theparentequation{eql-\the\eql@numbering@refcount}%
482     \protected@edef\theparentequation{\eql@tag@text}%
483   \else
484     \advance\c@equation@\@ne
485     \protected@edef\theparentequation{\theequation}%
486     \ifdefined\theHequation
487       \protected@edef\theparentequation{\theHequation}%
488     \fi

```

```

489 \fi
490 \global\c@parentequation\c@equation
491 \global\c@equation\z@
492 \let\theequation\eq@subequations@template
493 \def\theHequation{\theHparentequation.\arabic{equation}}%
494 }

```

`\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:

```

495 \def\eq@subequations@close{%
496 \ifnum\c@equation=\z@
497 \eq@subequations@restorecounter
498 \else
499 \global\c@equation\c@parentequation
500 \fi
501 }

```

`\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 `\eq@numbering@printsbeqlabel`

```

502 \def\eq@subequations@start{%
503 \let\eq@blocktag\@undefined
504 \let\eq@blocklabel\@undefined
505 \eq@nextopt@process{subequations}%
506 \eq@subequations@init
507 \global\advance\eq@numbering@refcount@\@ne
508 \edef\eq@subequations@currentHref{equation.eql-\the\eq@numbering@refcount@}%
509 \eq@Hy@anchor\eq@subequations@currentHref
510 \edef\eq@subequations@thepage{\thepage}%
511 \def\@currentcounter{equation}%
512 \let\@currentHref\eq@subequations@currentHref
513 \protected@edef\@currentlabel{\p@equation\theparentequation}%
514 \let\@currentlabelname\eq@labelname@default
515 \let\eq@subequations@active\eq@true
516 \ifdefined\eq@blocklabel
517 \ifdefined\eq@SK
518 \SK@\SK@@label\eq@blocklabel
519 \fi
520 \fi
521 \ignorespaces
522 }

```

`\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:

```

523 \def\eq@subequations@end{%
524 \ifnum\c@equation>\z@
525 \ifdefined\eq@blocklabel
526 \begingroup
527 \def\@currentcounter{equation}%
528 \let\thepage\eq@subequations@thepage
529 \let\@currentHref\eq@subequations@currentHref
530 % \TODO how about tag* ?! also within equations!
531 \protected@edef\@currentlabel{\p@equation\theparentequation}%

```

```

532     \let\@currentlabelname\eql@labelname@default
533     \expandafter\eql@label@clean\expandafter{\eql@blocklabel}%
534   \endgroup
535   \fi
536   \fi
537   \eql@subequations@close
538   \ignorespacesafterend
539 }

```

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

```

540 \newenvironment{eql@subequations}{%
541 (dev)\eql@dev@enterenv
542   \eql@subequations@testall\eql@subequations@start%
543 }{%
544   \eql@subequations@end
545 (dev)\eql@dev@leaveenv
546 }

```

**TODO:** describe

```

547 \def\eql@subequations@testall{\eql@subequations@testopt}
548 \def\eql@subequations@testopt#1{%
549   \eql@ifnextchar@tight [%]
550     {\eql@subequations@adopt{\eql@subequations@testat{#1}}}%
551     {\eql@subequations@testat{#1}}}
552 \def\eql@subequations@adopt#1[#2]{\eqnadopt{#2}#1}
553 \def\eql@subequations@testat#1{%
554   \eql@ifat@tight%
555     {\eql@subequations@addlabel{#1}}%
556     {#1}}
557 \def\eql@subequations@addlabel#1#2{\eqnadopt{label={#2}}#1}

```

## H Equation Numbering

**TODO:** describe

### H.1 Macros Shared with Amsmath

**TODO:** describe

`\maketag@@@`

```

558 \ifdefined\maketag@@@\else
559 \def\maketag@@@#1{\hbox{\m@th\normalfont#1}}
560 \fi

```

`\tagform@`

```

561 \ifdefined\tagform@\else
562 \def\tagform@#1{\maketag@@@{\ignorespaces#1\unskip\@italiccorr}}
563 \fi

564 \def\eql@tag@setmake@#1{%
565   \def\maketag@@@#1{\hbox{\m@th\normalfont#1}}%
566 }

```

```

567 \def\eql@tag@setmake#1{%
568   \def\maketag@@@#1{\hbox{\m@th\normalfont#1}}}%
569 }

```

**TODO:** describe

```

570 \def\eql@tag@setform@#1{%
571   \def\tagform@@#1{\maketag@@@{#1}}}%
572 }
573 \def\eql@tag@setform#1#2#3{%
574   \def\tagform@@#1{\maketag@@@{#1\ignorespaces#2\unskip\@italiccorr#3}}}%
575 }

```

**raise tags**

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

```

576 \newdimen\eql@raisetag@amount@

```

`\raisetag`

```

577 \def\eql@raisetag@default{%
578   \eql@warning{\string\raisetag\space not allowed here}
579   \@gobble
580 }

```

**TODO:** describe

```

581 \eql@amsmath@after{\let\eql@raisetag@default\raisetag}
582 \eql@amsmath@futurebefore{\let\raisetag\undefined}
583 \let\raisetag\eql@raisetag@default

```

**TODO:** maybe introduce a star form to enforce raise?

```

584 \def\eql@raisetag#1{\setlength\dimen@{#1}\global\eql@raisetag@amount@\dimen@}%

```

## H.2 Showkeys Integration

**TODO:** describe

```

585 \let\eql@SK\undefined
586 \let\eql@SK@lab@relax\@empty
587 \let\eql@SK@makelabel\@gobble
588 \let\eql@SK@printlabel@right\@empty
589 \let\eql@SK@printlabel@left\@empty
590 \let\eql@SK@printlabel@line\@empty
591 \def\eql@label@clean{\eql@label@org}
592 \AddToHook{package/showkeys/after}{
593   \let\eql@SK\@empty
594   \def\eql@SK@lab@relax{\let\eql@SK@lab@relax}
595   \eql@SK@lab@relax
596   \def\eql@SK@@label#1>#2\SK@{%
597     \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
598   }
599   \def\eql@SK@makelabel#1{%
600     \SK@\eql@SK@@label#1%
601   }
602   \def\eql@SK@printlabel@right{%
603     \ifx\eql@SK@lab@relax\else

```

```

604     \rlap{\kern\marginparsep\eqL@SK@lab}%
605     \eqL@SK@lab@relax
606   \fi
607 }
608 \def\eqL@SK@printlabel@left{%
609   \ifx\eqL@SK@lab\relax\else
610     \llap{\eqL@SK@lab\kern\marginparsep}%
611     \eqL@SK@lab@relax
612   \fi
613 }
614 \def\eqL@SK@printlabel@line{%
615   \ifx\eqL@SK@lab\relax\else
616     \dimen@\prevdepth
617     \nointerlineskip
618     \ifdefined\eqL@tagsleft
619       \llap{%
620         \eqL@SK@lab
621         \kern\marginparsep
622       }%
623     \eqL@SK@lab@relax
624   \else
625     \rlap{%
626       \dimen@\displaywidth
627       \advance\dimen@\marginparsep
628       \kern\dimen@
629       \eqL@SK@lab
630     }%
631   \fi
632   \eqL@SK@lab@relax
633   \prevdepth\dimen@
634 \fi
635 }
636 \let\eqL@label@org\label
637 \def\eqL@label@clean{\let\SK@\gobbletwo\eqL@label@org}
638 }

```

### H.3 Labels

**TODO:** describe

```

639 % \TODO implement (via label[] or labelname similar to label/tag)
640 \let\eqL@nextlabel\@undefined
641 \def\eqL@labelname@default{[equation]}

```

\eqL@label@org

```

642 \let\eqL@label@org\label

```

### H.4 Tags

**TODO:** describe

```

643 \let\eqL@nexttag\@undefined

```

\eqL@tag@default

```

644 \def\eqL@tag@default{%

```



```

645 \eql@error{\string\tag\space not allowed here}{}\eql@tag@gobble}
646 \eql@amsmath@after{
647 \let\eql@tag@default\tag
648 }
649 \let\tag\eql@tag@default

```

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

```

650 \def\eql@tag@gobble@[#1]#2{
651 \def\eql@tag@gobble{%
652 \eql@ampprotecttwo\eql@teststaropt@tight\eql@tag@gobble@\eql@tag@gobble@{}}

```

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

`\eql@tag@makenext` hyperref anchors  
`\eql@tag@makenext@@`  
`\eql@tag@makenext@@@`

```

653 \let\eql@Hy@anchor@\@gobble
654 \AddToHook{package/hyperref/after}{
655 \def\eql@Hy@anchor#1{%
656 \Hy@raisedlink{\hyper@anchor{#1}}%
657 }%
658 }

659 \def\eql@tag@makenext{%
660 \eql@ampprotecttwo\eql@teststaropt@tight
661 \eql@tag@makenext@star\eql@tag@makenext@\eql@tag@text
662 }

```

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

```

663 \def\eql@tag@makenext@star[#1]#2{%
664 \global\def\eql@nexttag{%
665 \let\eql@tag@tool\maketag@@@
666 \protected@edef\eql@tag@text{#2}%
667 \protected@edef\eql@tag@label{#1}%
668 }%
669 }

670 \def\eql@tag@makenext@[#1]#2{%
671 \global\def\eql@nexttag{%
672 \let\eql@tag@tool\tagform@
673 \protected@edef\eql@tag@text{#2}%
674 \protected@edef\eql@tag@label{#1}%
675 \protected@edef\eql@tag@label{\p@equation\eql@tag@label}%
676 }%
677 }

```

## H.5 Anchors

**TODO:** describe

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

```

678 \newcount\eql@numbering@refcount@
679 \eql@numbering@refcount@\z@

```

**TODO:** describe

```

680 \def\eql@numbering@makeblockanchor{%
681 \global\advance\eql@numbering@refcount@\@ne
682 \global\edef\eql@label@currentHref{equation.eql-\the\eql@numbering@refcount@}%

```

```

683 \eql@Hy@anchor\eql@label@currentHref
684 \global\edef\eql@label@thepage{\thepage}%
685 }
686 \def\eql@numbering@setblockanchor{%
687 \let\thepage\eql@label@thepage
688 \let\@currentHref\eql@label@currentHref
689 }

```

## H.6 Tag Composition

**TODO:** describe

```

\eql@compose@anchor
  \eql@compose@tag
  \eql@compose@label
690 \def\eql@compose@anchor{%
691 \ifdefined\eql@nexttag
692 \eql@nexttag
693 \def\@currentcounter{equation}%
694 \let\@currentlabel\eql@tag@label
695 \global\advance\eql@numbering@refcount\@ne
696 \edef\@currentHref{equation.eql-\the\eql@numbering@refcount}%
697 \eql@Hy@anchor\@currentHref
698 \global\let\eql@nexttag\@undefined
699 \else
700 \refstepcounter{equation}%
701 \let\eql@tag@tool\tagform@
702 \edef\eql@tag@text{\theequation}%
703 \fi
704 }

705 \def\eql@compose@label{%
706 \ifmeasuring@\else
707 \eql@SK@lab@relax
708 \ifdefined\eql@nextlabel
709 \ifnum
710 \ifnum\eql@numbering@target@<\z@
711 \eql@row@
712 \else
713 \eql@numbering@target@
714 \fi=\eql@row@
715 \eql@compose@label@
716 \fi
717 \fi
718 \fi
719 }

```

**TODO:** describe

```

720 \def\eql@compose@label@{%
721 \eql@SK@makelabel\eql@nextlabel
722 \begingroup
723 \ifnum\eql@numbering@target@=\eql@row@
724 \eql@numbering@setblockanchor
725 \fi
726 \let\@currentlabelname\eql@labelname@default
727 \expandafter\eql@label@clean\expandafter{\eql@nextlabel}%
728 \global\let\eql@nextlabel\@undefined
729 \endgroup
730 }

```

**TODO:** describe

```
731 \def\eql@compose@tag{%
732   \eql@tag@tool\eql@tag@text
733 }
```

**TODO:** describe

```
734 \def\eql@compose@print{%
735   \eql@compose@anchor
736   \eql@compose@label
737   \ifdefined\eql@tagsleft
738     \eql@SK@printlabel@left
739     \eql@compose@tag
740   \else
741     \eql@compose@tag
742     \eql@SK@printlabel@right
743   \fi
744 }
```

**TODO:** describe

```
745 \def\eql@compose@measure{%
746   \ifdefined\eql@nexttag
747     \eql@nexttag
748     \eql@tag@tool\eql@tag@text
749   \else
750     \stepcounter{equation}%
751     \tagform@theequation
752   \fi
753   \ifnum\eql@numbering@target@<\z@
754     \global\let\eql@nextlabel\@undefined
755     \global\let\eql@nexttag\@undefined
756   \fi
757 }
```

**TODO:** describe

```
758 \def\eql@compose@null{%
759   \ifdefined\eql@nexttag\else
760     \stepcounter{equation}%
761   \fi
762   \ifnum\eql@numbering@target@<\z@
763     \global\let\eql@nextlabel\@undefined
764     \global\let\eql@nexttag\@undefined
765   \fi
766 }
```

## H.7 Tagbox Methods

**TODO:** describe

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

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

774 }

**TODO:** describe

```
775 \def\eq\tagbox@print@right{%
776   \kern-\wd\eq\tagbox@
777   \box\eq\tagbox@
778 }
```

**TODO:** describe

```
779 \def\eq\tagbox@print@left{%
780   \wd\eq\tagbox@\z@
781   \box\eq\tagbox@
782 }
```

**TODO:** describe

```
783 \def\eq\tagbox@print@right@raise{%
784   \ifnum\eq\row@=\eq\totalrows@
785     \global\advance\eq\raisetag@firstlast@\tw@
786   \fi
787   \llap{\vtop{%
788     \vskip-\eq\raisetag@amount@
789     \normalbaselines
790     \setbox\@ne\hbox{}}%
791     \dp\@ne\eq\line@depth@
792     \box\@ne
793     \box\eq\tagbox@
794   }}%
795 }
796 \def\eq\tagbox@print@left@raise{%
797   \ifnum\eq\row@=\@one
798     \global\advance\eq\raisetag@firstlast@\@one
799   \fi
800   \rlap{\vbox{%
801     \normalbaselines
802     \box\eq\tagbox@
803     \vbox to\eq\line@height@{}}%
804     \vskip\eq\raisetag@amount@
805   }}%
806 }
```

**TODO:** describe

```
807 \def\eq\numbering@printsubeqlabel{%
808   \ifdefined\eq\parentlabel
809     \eq\numbering@makeblockanchor
810     \eq@SK@makelabel\eq\parentlabel
811     \begingroup
812       \def\@currentcounter{equation}%
813       \eq\numbering@setblockanchor
814       \let\@currentlabelname\eq\labelname@default
815       \protected@edef\@currentlabel{\p@equation\theparentequation}%
816       \expandafter\eq\label@clean\expandafter{\eq\parentlabel}%
817     \endgroup
818     \eq@SK@printlabel@line
819   \fi
820 }
```

## H.8 Numbering Schemes

**TODO:** describe

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

**TODO:** describe

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

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

850 \def\eq@numbering@set#1{%
851   \ifcsname eq@numbering@tab@#\endcsname
852     \expandafter\let\expandafter\eq@numbering@mode
853     \csname eq@numbering@tab@#\endcsname
854   \else
855     \eq@error{numbering mode ‘#1’ unknown: setting to ‘all’}%
856     \let\eq@numbering@mode\eq@numbering@tab@all
857   \fi
858 }
```

ing@target@ (*counter*)

```
859 \let\eq@numbering@active\eq@true
860 \newcount\eq@numbering@target@

861 \def\eq@numbering@mode@all{%
862   \eq@numbering@target@\m@ne}
863 \def\eq@numbering@mode@sub{%
864   \eq@numbering@target@\m@ne
865   \let\eq@numbering@subeq@use\eq@true}
866 \def\eq@numbering@mode@none{%
```

```

867 \eql@numbering@target@m@ne
868 \let\eql@numbering@active\eql@false}
869 \def\eql@numbering@mode@first{%
870 \eql@numbering@target@ne}
871 \def\eql@numbering@mode@last{%
872 \eql@numbering@target@MM}
873 \def\eql@numbering@mode@here{%
874 \eql@numbering@target@z}

```

**TODO:** describe

```

875 \def\eql@numbering@mode@in{%
876 \ifdefined\eql@tagsleft
877 \eql@numbering@mode@last
878 \else
879 \eql@numbering@mode@first
880 \fi
881 }

```

**TODO:** describe

```

882 \def\eql@numbering@mode@out{%
883 \ifdefined\eql@tagsleft
884 \eql@numbering@mode@first
885 \else
886 \eql@numbering@mode@last
887 \fi
888 }

```

**TODO:** describe

```

889 \def\eql@numbering@mode@middle{%
890 \eql@numbering@target@z}
891 \let\eql@numbering@eval@target\eql@numbering@eval@middle}
892 \def\eql@numbering@eval@middle{%
893 \ifnum\eql@numbering@target@=z}
894 \count@eql@row@
895 \advance\count@ne
896 \divide\count@tw@
897 \global\eql@numbering@target@\count@
898 \fi
899 }

```

**TODO:** describe

```

900 \def\eql@numbering@eval@mode{%
901 \let\eql@numbering@eval@target@undefined
902 \let\eql@numbering@subeq@use\eql@false
903 \csname eql@numbering@mode@\eql@numbering@mode@endcsname
904 \ifdefined\eql@numbering@active
905 \let\eql@numbering@eqnswinit@eqnswtrue
906 \else
907 \let\eql@numbering@eqnswinit@eqnswfalse
908 \fi
909 \let\eql@numbering@active\eql@false
910 }

```

**TODO:** reconsider operation

\numberhere

```

911 \def\numberhere{%
912   \ifmeasuring@
913     \ifnum\eq@numbering@target@<\z@\else
914       \global\eq@numbering@target@\eq@row@
915     \fi
916   \fi
917 }

```

**TODO:** describe

`\numbernext`

```

918 \def\numbernext{%
919   \ifmeasuring@
920     \ifnum\eq@numbering@target@<\z@\else
921       \ifnum\eq@numbering@target@=\eq@row@
922         \global\advance\eq@numbering@target@\@ne
923       \fi
924     \fi
925   \fi
926 }

```

## H.9 Numbering Framework

**TODO:** describe

```

927 \let\eq@numbering@autolabel\eq@false
928 \let\eq@numbering@autotag\eq@true
929 \let\eq@numbering@blocklabel\@undefined
930 \let\eq@numbering@blocktag\@undefined

```

**TODO:** where does this belong?

```

931 \eq@amsmath@after{
932   \let\eq@print@eqnum@default\print@eqnum
933   \let\eq@incr@eqnum@default\incr@eqnum
934 }

```

**TODO:** describe

```

935 \def\donumber{%
936   \if@eqnsw\else
937     \global\@eqnswtrue
938     \ifx\print@eqn\@empty
939       \global\let\print@eqn\eq@print@eqnum@default
940     \fi
941     \ifx\incr@eqn\@empty
942       \global\let\incr@eqn\eq@incr@eqnum@default
943     \fi
944   \fi
945 }

```

**TODO:** describe

```

946 \def\eq@label@warn{%
947   \ifdefined\eq@numbering@autolabel
948     \global\@eqnswtrue
949   \fi
950   \ifdefined\eq@nextlabel
951     \eq@warn@label@multiple\eq@nextlabel

```

```
952 \fi
953 \global\edef\eql@nextlabel
954 }
```

**TODO:** describe

```
955 \def\eql@tag@warn{%
956 \ifdefined\eql@numbering@autotag
957 \global\@eqnswtrue
958 \fi
959 \ifdefined\eql@nexttag
960 \eql@warn@tag@multiple
961 \fi
962 \eql@tag@makenext
963 }
```

**TODO:** describe

```
964 \def\eql@label@nowarn{%
965 \ifdefined\eql@numbering@autolabel
966 \global\@eqnswtrue
967 \fi
968 \global\edef\eql@nextlabel
969 }
```

**TODO:** describe

```
970 \def\eql@tag@nowarn{%
971 \ifdefined\eql@numbering@autotag
972 \global\@eqnswtrue
973 \fi
974 \eql@tag@makenext
975 }
```

**TODO:** describe

```
976 \def\eql@blocklabel@set#1{%
977 \ifdefined\eql@blocklabel
978 \eql@warn@label@multiple\eql@blocklabel
979 \fi
980 \edef\eql@blocklabel{#1}%
981 }
```

**TODO:** describe

```
982 \def\eql@blocktag@set#1{%
983 \ifdefined\eql@blocktag
984 \eql@warn@tag@multiple
985 \fi
986 \def\eql@blocktag{#1}%
987 }
```

**TODO:** describe

```
988 \def\eql@blocktag@setstar#1{%
989 \ifdefined\eql@blocktag
990 \eql@warn@tag@multiple
991 \fi
992 \def\eql@blocktag{*#1}%
993 }
```



**Single-Line** **TODO:** describe

```
994 \def\eq@numbering@single@init{%
995   \let\label\eq@label@warn
996   \let\tag\eq@tag@warn
997   \let\raisetag\eq@raisetag
998   \eq@numbering@target@{m@ne
999   \let\eq@nextlabel\eq@blocklabel
1000  \ifdefined\eq@blocktag
1001    \expandafter\eq@tag@makenext\eq@blocktag
1002  \else
1003    \let\eq@nexttag\@undefined
1004  \fi
1005  \eq@numbering@eqswinit
1006  \ifdefined\eq@numbering@autolabel
1007    \ifdefined\eq@nextlabel
1008      \eqswtrue
1009    \fi
1010  \fi
1011  \ifdefined\eq@numbering@autotag
1012    \ifdefined\eq@nexttag
1013      \eqswtrue
1014    \fi
1015  \fi
1016  \global\eq@raisetag@amount@{z@
1017 }
```

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

```
1018 \def\eq@numbering@measure@init{%
1019   \let\label\eq@label@warn
1020   \let\tag\eq@tag@warn
1021   \let\raisetag\eq@raisetag
1022   \global\let\eq@nextlabel\eq@blocklabel
1023   \ifdefined\eq@blocktag
1024     \expandafter\eq@tag@makenext\eq@blocktag
1025   \else
1026     \global\let\eq@nexttag\@undefined
1027   \fi
1028   \ifnum\eq@numbering@target@<{z@\else
1029     \eq@numbering@eqswinit
1030     \ifdefined\eq@numbering@autolabel
1031       \ifdefined\eq@nextlabel
1032         \eqswtrue
1033       \fi
1034     \fi
1035   \fi
1036 }
```

**TODO:** describe

```
1037 \def\eq@numbering@measure@line@begin{%
1038   \ifnum\eq@numbering@target@<{z@
1039     \global\eq@numbering@eqswinit
1040   \fi
1041 }
```

**TODO:** describe

```

1042 \def\eq@numbering@measure@eval{%
1043   \ifdefined\eq@numbering@eval@target
1044     \eq@numbering@eval@target
1045   \fi
1046   \ifnum\eq@numbering@target@>\eq@row@
1047     \global\eq@numbering@target@\eq@row@
1048   \fi
1049   \ifnum\eq@numbering@target@>\z@
1050     \if@eqnsw\else
1051       \global\eq@numbering@target@\z@
1052     \fi
1053   \fi
1054   \ifnum\eq@numbering@target@<\@ne
1055     \ifdefined\eq@nextlabel
1056       \eq@warn@label@unused
1057       \global\let\eq@nextlabel\@undefined
1058     \fi
1059     \ifdefined\eq@nexttag
1060       \eq@warn@tag@unused
1061       \global\let\eq@nexttag\@undefined
1062     \fi
1063   \fi
1064 }

```

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

```

1065 \def\eq@numbering@print@init{%
1066   \ifnum\eq@numbering@target@<\z@
1067     \let\label\eq@label@nowarn
1068     \let\tag\eq@tag@nowarn
1069     \let\raisetag\eq@raisetag
1070     \let\eq@nextlabel\eq@blocklabel
1071     \ifdefined\eq@blocktag
1072       \expandafter\eq@tag@makenext\eq@blocktag
1073     \else
1074       \let\eq@nexttag\@undefined
1075     \fi
1076   \else
1077     \let\label\@gobble
1078     \let\tag\eq@tag@gobble
1079     \let\raisetag\eq@gobble
1080   \fi
1081 }

```

**TODO:** describe

```

1082 \def\eq@numbering@print@block@begin{%
1083   \ifnum\eq@numbering@target@>\z@
1084     \eq@numbering@makeblockanchor
1085   \fi
1086   \ifdefined\eq@numbering@subeq@use
1087     \eq@numbering@printsubeqlabel
1088   \fi
1089 }

```

**TODO:** describe

```

1090 \def\eq@numbering@print@line@begin{%
1091   \ifnum\eq@numbering@target@<\z@

```

```

1092   \global\eql@numbering@eqnswinit
1093   \global\eql@raisetag@amount@z@
1094   \fi
1095 }

```

**TODO:** describe

```

1096 \def\eql@numbering@print@line@eval{%
1097   \ifnum\eql@numbering@target@<z@\else
1098     \ifnum\eql@numbering@target@=\eql@row@
1099       \global\@eqnswtrue
1100     \else
1101       \global\@eqnswfalse
1102     \fi
1103   \fi
1104 }

```

## H.10 Subequations

**TODO:** describe

```

1105 \def\eql@numbering@subeq@init{%
1106   \let\eql@save@theequation\theequation
1107   \let\eql@save@theHequation\theHequation
1108   \eql@subequations@init
1109   \let\eql@parentlabel\eql@blocklabel
1110   \let\eql@parenttag\eql@blocktag
1111   \let\eql@blocklabel\@undefined
1112   \let\eql@blocktag\@undefined
1113 }

```

**TODO:** describe

```

1114 \def\eql@numbering@subeq@test{%
1115   \ifnum\c@equation<\tw@
1116     \let\eql@numbering@subeq@use\@ne
1117   \fi
1118 }

```

**TODO:** describe

```

1119 \def\eql@numbering@subeq@revert{%
1120   \let\eql@blocklabel\eql@parentlabel
1121   \let\eql@blocktag\eql@parenttag
1122   \let\eql@numbering@subeq@use\eql@false
1123   \let\theequation\eql@save@theequation
1124   \let\theHequation\eql@save@theHequation
1125   \eql@subequations@restorecounter
1126 }

```

**TODO:** describe

```

1127 % \TODO note must not use setcounter here (when calc is loaded)
1128 \def\eql@numbering@subeq@close{%
1129   \eql@subequations@close
1130 }

```

# I Display Equations Support

**TODO:** describe

## I.1 Display Breaks

**TODO:** describe

erdisplaylinepenalty

```
1131 \interdisplaylinepenalty\@M
```

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

```
1132 \def\eql@getdsp@pen#1{%
1133   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1134 }
```

**TODO:** allow a displaybreak before equations

```
1135 \DeclareRobustCommand{\eql@displaybreak@default}[1][4]{%
1136   \eql@warning{Invalid use of \string\displaybreak-}}
1137 \eql@amsmath@futurebefore{\let\displaybreak\@undefined}
1138 \eql@amsmath@after{\let\eql@displaybreak@default\displaybreak}
1139 \let\displaybreak\eql@displaybreak@default
```

```
1140 \newcount\eql@displaybreak@pen@
1141 \newcount\eql@displaybreak@prepen@
```

```
1142 \protected\def\eql@displaybreak@print{%
1143   \eql@ampprotect\eql@testopt@tight\eql@displaybreak@print@{4}%
1144 }
```

**TODO:** describe

```
1145 \def\eql@displaybreak@print@[#1]{%
1146   \ifnum#1<\z@
1147     \global\eql@displaybreak@pen@\@MM
1148   \else
1149     \global\eql@displaybreak@pen@-\@getpen{#1}\relax
1150   \fi
1151 }
```

**TODO:** describe

```
1152 \def\eql@displaybreak@pre#1{%
1153   \ifnum#1<\z@
1154     \global\eql@displaybreak@prepen@\@MM
1155   \else
1156     \global\eql@displaybreak@prepen@-\@getpen{#1}\relax
1157   \fi
1158 }
```

**TODO:** describe

```
1159 \protected\def\eql@displaybreak@measure{%
1160   \eql@ampprotect\eql@testopt@tight\eql@displaybreak@measure@{4}%
1161 }
1162 \def\eql@displaybreak@measure@[#1]{}
```

## I.2 General Initialisation

**TODO:** describe

`\eql@display@atend`

```
1163 \def\eql@display@atend#1{%
1164   \expandafter\def\expandafter\eql@display@doatend\expandafter
1165   {\eql@display@doatend#1}%
1166 }
```

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

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

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

```
1167 \newskip\eql@vspaceskip@
1168 \newskip\eql@abovespace@
1169 \newskip\eql@belowspace@
1170 \let\eql@vspace@org\vspace
1171 \def\eql@vspace{\eql@ifstar@loose\eql@vspace@\eql@vspace@}
1172 \def\eql@vspace#1{%
1173   \setlength\skip@{#1}%
1174   \global\advance\eql@vspaceskip@\skip@
1175 }
```

`\eql@display@init`

```
1176 \def\eql@display@init{%
1177   \eql@display@firstavail@z@
1178   \eql@raisetag@firstlast@z@
1179   \let\displaybreak\eql@displaybreak@print
1180   \eql@displaybreak@pen@\@MM
1181   \eql@vspaceskip@z@skip
1182   \let\eql@vspace@org\vspace
1183   \let\vspace\eql@vspace
1184 }
```

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

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

```
1185 \def\eql@display@close{%
1186   \ifdim\eql@halign@prevdepth@=\maxdimen
1187     \ifdim\predisplaysize@=\maxdimen
1188     % \TODO hbox was clear (noindent)
1189     \abovedisplayskip\eql@abovedisplaycontskip@
1190     % \TODO whether to enable or disable short
1191     \predisplaysize\maxdimen
1192     \abovedisplayshortskip\abovedisplayskip
1193     \belowdisplayshortskip\belowdisplayskip
1194     \fi
1195   \else
1196   % \TODO should tagskip be allowed to override these?! rather not
1197     \ifdim\eql@halign@prevdepth@=\@m\p@
1198     \displaypenaltyz@
1199     \abovedisplayskip\eql@abovedisplaytopskip@
1200     % \TODO whether to offer this?
1201     \belowdisplayskip\eql@belowdisplaytopskip@
1202     \else
1203     \displaypenaltyz@
1204     \abovedisplayskip\eql@abovedisplayparskip@
1205     \fi

```

```

1206 % \TODO whether to enable or disable short
1207   \predisplaysize\maxdimen
1208   \abovedisplayshortskip\abovedisplayskip
1209   \belowdisplayshortskip\belowdisplayskip
1210 \fi
1211 \eql@display@doatend
1212 \ifdim\eql@display@firstavail@<\z@
1213   \eql@display@firstavail@\z@
1214 \fi
1215 \advance\eql@display@firstavail@\displayindent
1216 \ifdim\eql@display@firstavail@>\predisplaysize
1217   \ifcase\eql@displayshortmode@
1218     \or
1219       \abovedisplayskip\abovedisplayshortskip
1220     \or
1221       \abovedisplayskip\abovedisplayshortskip
1222       \ifnum\eql@row@=\tw@
1223         \belowdisplayskip\belowdisplayshortskip
1224       \fi
1225     \or
1226       \abovedisplayskip\abovedisplayshortskip
1227       \belowdisplayskip\belowdisplayshortskip
1228     \fi
1229 \fi
1230 \ifodd\eql@raisetag@firstlast@
1231   \abovedisplayskip\eql@abovedisplaytagskip@
1232 \fi
1233 \ifnum\eql@raisetag@firstlast@>\@ne
1234   \belowdisplayskip\eql@belowdisplaytagskip@
1235 \fi
1236 \ifnum\eql@displaybreak@pen@=\@MM\else
1237   \postdisplaypenalty\eql@displaybreak@pen@
1238 \fi
1239 \ifnum\eql@displaybreak@prepen@=\@MM\else
1240   \predisplaypenalty\eql@displaybreak@prepen@
1241 \fi
1242 % \TODO here or above?
1243 \advance\abovedisplayskip\eql@abovespace@
1244 \advance\belowdisplayskip\eql@belowspace@
1245 \advance\belowdisplayskip\eql@vspaceskip@
1246 \count@\prevgraf
1247 \advance\count@\eql@row@
1248 \advance\count@-\tw@
1249 \prevgraf\count@
1250 }

1251 \def\eql@display@leave{%
1252   \let\label\eql@label@org
1253   \let>tag\eql@tag@default
1254   \let\raisetag\eql@raisetag@default
1255   \let\displaybreak\eql@displaybreak@default
1256   \let\vspace\eql@vspace@org
1257 }

1258 \expandafter\def\expandafter\@arrayparboxrestore\expandafter{%
1259   \@arrayparboxrestore
1260   \eql@display@leave
1261 % \TODO should we always revert to originals? or only if switch is set
1262 \eql@amprevert
1263 \@displayfalse

```

1264 }

### I.3 halign Support

**TODO:** describe

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

```
1265 \newbox\eql@strutbox@
1266 \def\eql@strut{\copy\eql@strutbox@}
1267 \let\eql@strut@field\eql@strut
1268 \let\eql@strut@tag\eql@strut
1269 \def\eql@strut@make{%
1270   \setbox\eql@strutbox@\hbox{%
1271     \@tempdima\normalbaselineskip
1272     \advance\@tempdima-\normallineskiplimit
1273     \@tempdimb.3\normalbaselineskip
1274     \advance\@tempdimb.5\normallineskiplimit
1275     \advance\@tempdima-\@tempdimb
1276     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1277   }
1278 }
1279 \AtBeginDocument{\eql@strut@make}
```

**TODO:** describe

```
1280 \def\eql@halign@spread{%
1281   \dimen@\eql@spread\relax
1282   \advance\dimen@\normalbaselineskip
1283   \ifdim\dimen@>\baselineskip
1284     \normalbaselines
1285     \dimen@\eql@spread\relax
1286     \advance\dimen@\normalbaselineskip
1287     \advance\dimen@-\baselineskip
1288     \openup\dimen@
1289     \ifdefined\spread@equation
1290       \let\spread@equation\@empty
1291     \fi
1292   \fi
1293 }
```

`gn@prevdepth@` (*dimen*)

```
1294 \newdimen\eql@halign@prevdepth@
1295 \def\eql@halign@catchprevdepth{%
1296   \ifvmode
1297     \eql@halign@prevdepth@\prevdepth
1298     \nointerlineskip
1299     \noindent
1300   \else
1301     \eql@halign@prevdepth@\maxdimen
1302   \fi
1303 }

1304 \def\eql@halign@before{%
1305   \ifdim\eql@halign@prevdepth@=\maxdimen\else
1306     \prevdepth\eql@halign@prevdepth@
```

```

1307 \fi
1308 \ifdim\prevdepth=-\@m\p\else
1309   \skip@\normalbaselineskip
1310   \advance\skip@-\eql@display@height\relax
1311   \advance\skip@-\prevdepth\relax
1312   \ifdim\skip@<\normallineskiplimit
1313     \vskip\normallineskip
1314   \else
1315     \vskip\skip@
1316   \fi
1317   \nointerlineskip
1318 \fi
1319 }

```

**TODO:** describe

```

1320 \def\eql@halign@after{%
1321   \prevdepth\eql@display@depth\relax
1322 }

```

**TODO:** describe

```

1323 \def\eql@halign@init#1{%
1324   \eql@halign@spread
1325   \eql@strut@make
1326   \everycr{\noalign{#1}}%
1327 }

```

## I.4 Stack

**TODO:** describe

```

1328 \def\eql@stack@enable{%
1329   \let\eql@stack@save@single\eql@stack@save@single@
1330   \let\eql@stack@save@multi\eql@stack@save@multi@
1331   \let\eql@stack@save@boxed\eql@stack@save@boxed@
1332 }

```

**TODO:** describe

```

1333 \let\eql@stack@save@single\eql@stack@enable
1334 \let\eql@stack@save@multi\eql@stack@enable
1335 \let\eql@stack@save@boxed\eql@stack@enable
1336 \let\eql@stack@restore@empty

```

**TODO:** describe

```

1337 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
1338 \def\eql@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

**TODO:** describe

```

1339 \def\eql@stack@save@single@{%
1340   \let\eql@stack@nextlabel\eql@nextlabel
1341   \let\eql@stack@nexttag\eql@nexttag
1342   \edef\eql@stack@restore{%
1343     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1344     \eql@stack@save@let\eql@stack@nextlabel\eql@nextlabel
1345     \eql@stack@save@let\eql@stack@nexttag\eql@nexttag
1346     \eql@stack@save@reg\eql@displaybreak@pen@

```



```

1347 \eql@stack@save@reg\eql@vspaceskip@
1348 \eql@stack@save@reg\eql@shape@pos@
1349 \eql@stack@save@reg\eql@shape@amount@
1350 \eql@stack@save@reg\eql@display@firstavail@
1351 \eql@stack@save@reg\eql@raisetag@amount@
1352 \eql@stack@save@reg\eql@raisetag@firstlast@
1353 }%
1354 }

```

**TODO:** describe

```

1355 \def\eql@stack@save@multi@{%
1356 \let\eql@stack@nextlabel\eql@nextlabel
1357 \let\eql@stack@nexttag\eql@nexttag
1358 \let\eql@stack@tagwidth@tab\eql@tagwidth@tab
1359 \let\eql@stack@fieldlength@tab\eql@fieldlength@tab
1360 \let\eql@stack@colwidth@tab\eql@colwidth@tab
1361 \let\eql@stack@label@thepage\eql@label@thepage
1362 \let\eql@stack@label@currentHref\eql@label@currentHref
1363 \edef\eql@stack@restore{%
1364 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1365 \eql@stack@save@let\eql@stack@nextlabel\eql@nextlabel
1366 \eql@stack@save@let\eql@stack@nexttag\eql@nexttag
1367 \eql@stack@save@let\eql@stack@tagwidth@tab\eql@tagwidth@tab
1368 \eql@stack@save@let\eql@stack@fieldlength@tab\eql@fieldlength@tab
1369 \eql@stack@save@let\eql@stack@colwidth@tab\eql@colwidth@tab
1370 \eql@stack@save@let\eql@stack@label@thepage\eql@label@thepage
1371 \eql@stack@save@let\eql@stack@label@currentHref\eql@label@currentHref
1372 \eql@stack@save@reg\eql@displaybreak@open@
1373 \eql@stack@save@reg\eql@vspaceskip@
1374 \eql@stack@save@reg\eql@shape@pos@
1375 \eql@stack@save@reg\eql@shape@amount@
1376 \eql@stack@save@reg\eql@display@firstavail@
1377 \eql@stack@save@reg\eql@raisetag@amount@
1378 \eql@stack@save@reg\eql@raisetag@firstlast@
1379 \eql@stack@save@reg\eql@column@
1380 \eql@stack@save@reg\eql@totalcolumns@
1381 \eql@stack@save@reg\eql@line@avail@
1382 \eql@stack@save@reg\eql@line@pos@
1383 \eql@stack@save@reg\eql@line@width@
1384 \eql@stack@save@reg\eql@line@depth@
1385 \eql@stack@save@reg\eql@line@height@
1386 \eql@stack@save@reg\eql@numbering@target@
1387 \eql@stack@save@reg\eql@row@
1388 }%
1389 }
1390 \def\eql@stack@save@boxed@{%
1391 \edef\eql@stack@restore{%
1392 \eql@stack@save@reg\eql@row@
1393 \eql@stack@save@reg\eql@totalrows@
1394 \eql@stack@save@reg\eql@shape@pos@
1395 \eql@stack@save@reg\eql@shape@amount@
1396 }%
1397 }

```

## J 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.

### J.1 Supporting Definitionss

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

```
1398 \ifdefined\inf@bad\else%
1399   \newcount\inf@bad
1400   \inf@bad1000000\relax
1401 \fi
```

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

```
1402 \let\eql@restore@hfuzz\@empty
1403 \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:

```
1404 \newcount\eql@shape@pos@
1405 \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:

```
1406 \newdimen\eql@marginleft@
1407 \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:

```
1408 \newcount\eql@marginbadness@
1409 \newcount\eql@maxbadness@
1410 \eql@marginbadness@\inf@bad
1411 \eql@maxbadness@\inf@bad
```

### J.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:

```
1412 \def\eql@shape@tab@default{default}
1413 \def\eql@shape@tab@left{left}
1414 \def\eql@shape@tab@center{center}
1415 \def\eql@shape@tab@right{right}
1416 \def\eql@shape@tab@first{first}
1417 \def\eql@shape@tab@hanging{hanging}
1418 \def\eql@shape@tab@steps{steps}
```

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

```

1419 \let\eq@shape@tab@def\eq@shape@tab@default
1420 \let\eq@shape@tab@\eq@shape@tab@default
1421 \let\eq@shape@tab@l\eq@shape@tab@left
1422 \let\eq@shape@tab@c\eq@shape@tab@center
1423 \let\eq@shape@tab@r\eq@shape@tab@right
1424 \let\eq@shape@tab@rc\eq@shape@tab@first
1425 \let\eq@shape@tab@indent\eq@shape@tab@first
1426 \let\eq@shape@tab@hang\eq@shape@tab@hanging
1427 \let\eq@shape@tab@lc\eq@shape@tab@hanging
1428 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
1429 \let\eq@shape@tab@lcr\eq@shape@tab@steps

```

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

```

1430 \let\eq@shape@mode\eq@shape@tab@default

```

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

```

1431 \def\eq@shape@set#1{%
1432   \ifcsname eq@shape@tab@#1\endcsname
1433     \expandafter\let\expandafter\eq@shape@mode
1434     \csname eq@shape@tab@#1\endcsname
1435   \else
1436     \eq@error{shape '#1' unknown: setting to default}%
1437     \let\eq@shape@mode\eq@shape@tab@default
1438   \fi
1439 }

```

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

```

1440 \def\eq@shape@center@left{\eq@shape@pos@z@\eq@shape@amount@z@}
1441 \def\eq@shape@center@center{\eq@shape@pos@ne\eq@shape@amount@z@}
1442 \def\eq@shape@center@right{\eq@shape@pos@tw@\eq@shape@amount@z@}
1443 \let\eq@shape@center@default\eq@shape@center@center
1444 \def\eq@shape@left@left{\eq@shape@pos@z@\eq@shape@amount@z@}
1445 \def\eq@shape@left@center{\eq@shape@pos@ne\eq@shape@amount@z@}
1446 \def\eq@shape@left@right{\eq@shape@pos@tw@\eq@shape@amount@z@}
1447 \let\eq@shape@left@default\eq@shape@left@left

```

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

```

1448 \def\eq@shape@center@first{%
1449   \eq@shape@pos@z@
1450   \eq@shape@amount@z@
1451   \ifnum\eq@totalrows@>\@ne
1452     \ifnum\eq@row@=\@ne
1453       \eq@shape@amount@\eq@indent@
1454     \fi
1455   \fi
1456 }
1457 \def\eq@shape@left@first{%
1458   \eq@shape@pos@z@
1459   \eq@shape@amount@z@
1460   \ifnum\eq@totalrows@>\@ne
1461     \ifnum\eq@row@=\@ne

```

```

1462     \eql@shape@amount@\eql@indent@
1463     \fi
1464 \fi
1465 }

```

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:

```

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

```

The `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:

```

1484 \def\eql@shape@center@steps{%
1485   \eql@shape@amount@\z@
1486   \eql@shape@pos@\@ne
1487   \ifnum\eql@totalrows@>\@ne
1488     \ifnum\eql@row@=\@ne
1489       \eql@shape@pos@\z@
1490     \fi
1491     \ifnum\eql@row@=\eql@totalrows@
1492       \eql@shape@pos@\tw@
1493     \fi
1494 \fi
1495 }
1496 \def\eql@shape@left@steps{%
1497   \eql@shape@pos@\z@
1498   \eql@shape@amount@\z@
1499   \ifnum\eql@totalrows@>\@ne
1500     \ifnum\eql@row@=\@ne
1501       \eql@shape@amount@-\eql@indent@
1502     \fi
1503     \ifnum\eql@row@=\eql@totalrows@
1504       \eql@shape@amount@\eql@indent@
1505     \fi
1506 \fi
1507 }

```

`\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`:

```

1508 \let\eq@shape@eval\@undefined
1509 \def\eq@shape@sel{%
1510   \expandafter\let\expandafter\eq@shape@eval
1511     \csname eq@shape%
1512     @\ifdefined\eq@flushleft left\else center\fi
1513     @\eq@shape@mode\endcsname
1514 }

```

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

`\eq@adjust@shoveright`

`\eq@adjust@shovecenter`

```

1515 \def\eq@adjust@shoveleft{%
1516   \global\eq@shape@pos@z@
1517   \eq@srbgroup\eq@ifstar@tight
1518     {\eq@adjust@shoveleft@[\eq@indent@]}%
1519     {\eq@ifnextgobble@tight{!}}%
1520     {\eq@adjust@shoveleft@[-\eq@indent@]}%
1521     {\eq@testopt@tight\eq@adjust@shoveleft@z@}%
1522   }%
1523 }
1524 \def\eq@adjust@shoveleft@[#1]{%
1525   \eq@sregroup\setlength\dimen@{#1}\global\eq@shape@amount@\dimen@
1526 \def\eq@adjust@shovecenter{%
1527   \global\eq@shape@pos@one\global\eq@shape@amount@z@
1528 \def\eq@adjust@shoveright{%
1529   \global\eq@shape@pos@tw@\global\eq@shape@amount@z@

```

### J.3 Adjustment Methods

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

```

1530 \def\eq@adjust@try#1{%
1531   \setbox\eq@fieldbox@\hbox to\displaywidth{%
1532     \unhbox\eq@fieldbox@\unkern\kern#1}%
1533 }

```

`\eq@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`:

```

1534 \def\eq@adjust@print#1#2{%
1535   \eq@restore@hfuzz
1536   \hbox to\displaywidth{#1\unhbox\eq@fieldbox@\unkern#2}%
1537 }

```

`\eq@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 `\eq@display@firstavail@` to the minimum left available space we can guarantee:

`\eq@adjust@print@aligncenter`

`\eq@adjust@print@alignright`

```

1538 \def\eq@adjust@print@alignleft#1#2{%
1539   \ifnum\eq@row@=\@one
1540     \global\eq@display@firstavail@#1%
1541     \fi
1542   \eq@adjust@print{\kern#1}{\kern#2}%
1543 }
1544 \def\eq@adjust@print@alignright#1#2{%
1545   \ifnum\eq@row@=\@one

```

```

1546 \eql@display@firstavail@\displaywidth
1547 \advance\eql@display@firstavail@-\eql@fieldwidth@
1548 \global\advance\eql@display@firstavail@-#2%
1549 \fi
1550 \eql@adjust@print{\kern#1\hfil}{\unskip\kern#2}%
1551 }
1552 \def\eql@adjust@print@aligncenter#1#2{%
1553 \ifnum\eql@row@=\@ne
1554 \eql@display@firstavail@\displaywidth
1555 \advance\eql@display@firstavail@-\eql@fieldwidth@
1556 \advance\eql@display@firstavail@#1%
1557 \advance\eql@display@firstavail@-#2%
1558 \global\divide\eql@display@firstavail@\tw@
1559 \fi
1560 \eql@adjust@print{\kern#1\hfil}{\kern#2}%
1561 }

```

`\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:

```

1562 \def\eql@adjust@init{%
1563 \eql@save@hfuzz
1564 \hfuzz\maxdimen
1565 \eql@shape@sel
1566 }

```

`\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:

```

1567 \def\eql@adjust@sel@tag{%
1568 \csname eql@adjust%
1569 @\ifdefined\eql@flushleft flushleft\else center\fi
1570 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1571 @\ifcase\eql@shape@pos@ shoveleft\or shovecenter\or shoveright\fi
1572 @tag\endcsname
1573 }
1574 \def\eql@adjust@sel@notag{%
1575 \csname eql@adjust%
1576 @\ifdefined\eql@flushleft flushleft\else center\fi
1577 @\ifdefined\eql@tagsleft tagsleft\else tagsright\fi
1578 @\ifcase\eql@shape@pos@ shoveleft\or shovecenter\or shoveright\fi
1579 @notag\endcsname
1580 }

```

## J.4 Centered Layout

**TODO:** describe

**TODO:** check all these!!

```

1581 \def\eql@adjust@center@tagsright@shovecenter@notag{%
1582 \dimen@\displaywidth
1583 \advance\dimen@-\eql@fieldwidth@
1584 \ifdim\dimen@>\eql@tagmargin@
1585 \eql@adjust@print@aligncenter\z@\eql@tagmargin@
1586 \else
1587 \eql@adjust@print@alignleft\z@\z@

```

```
1588 \fi
1589 }
```

**TODO:** describe

```
1590 \def\eq@adjust@center@tagsright@shovecenter@tag{%
1591   \dimen@\displaywidth
1592   \ifdim\eq@tagwidth@<\eq@tagmargin@
1593     \advance\dimen@-\eq@tagmargin@
1594   \else
1595     \advance\dimen@-2\eq@tagwidth@
1596     \advance\dimen@\eq@tagmargin@
1597   \fi
1598   \ifdim\eq@fieldwidth@<\dimen@
1599     \eq@adjust@print@aligncenter\z@\eq@tagmargin@
1600     \eq@tagbox@print@right
1601   \else
1602     \eq@adjust@try\eq@tagwidth@
1603     \ifnum\badness<\eq@maxbadness@
1604       \ifdim\eq@tagwidth@<\eq@tagmargin@
1605         \eq@adjust@print@alignleft\z@\eq@tagwidth@
1606       \else
1607         \eq@adjust@print@alignright\z@\eq@tagwidth@
1608       \fi
1609     \eq@tagbox@print@right
1610   \else
1611     \eq@adjust@center@tagsright@shovecenter@notag
1612     \eq@tagbox@print@right@raise
1613   \fi
1614 \fi
1615 }
```

**TODO:** describe

```
1616 \def\eq@adjust@center@tagsleft@shovecenter@notag{%
1617   \dimen@\displaywidth
1618   \advance\dimen@-\eq@tagmargin@
1619   \ifdim\eq@fieldwidth@<\dimen@
1620     \eq@adjust@print@aligncenter\eq@tagmargin@\z@
1621   \else
1622     \eq@adjust@print@alignright\z@\z@
1623   \fi
1624 }
```

**TODO:** describe

```
1625 \def\eq@adjust@center@tagsleft@shovecenter@tag{%
1626   \dimen@\displaywidth
1627   \ifdim\eq@tagwidth@<\eq@tagmargin@
1628     \advance\dimen@-\eq@tagmargin@
1629   \else
1630     \advance\dimen@-2\eq@tagwidth@
1631     \advance\dimen@\eq@tagmargin@
1632   \fi
1633   \ifdim\eq@fieldwidth@<\dimen@
1634     \eq@tagbox@print@left
1635     \eq@adjust@print@aligncenter\eq@tagmargin@\z@
1636   \else
1637     \eq@adjust@try\eq@tagwidth@
1638     \ifnum\badness<\eq@maxbadness@
1639       \eq@tagbox@print@left
```

```

1640     \ifdim\eqL@tagwidth@<\eqL@tagmargin@
1641     \eqL@adjust@print@alignright\eqL@tagwidth@z@
1642     \else
1643     \eqL@adjust@print@alignleft\eqL@tagwidth@z@
1644     \fi
1645     \else
1646     \eqL@tagbox@print@left@raise
1647     \eqL@adjust@center@tagsleft@shovecenter@notag
1648     \fi
1649     \fi
1650     \eqL@display@firstavail@set\z@
1651 }

```

**TODO:** describe

```

1652 \def\eqL@adjust@center@tagsright@shoveleft@notag{%
1653   \dimen@displaywidth
1654   \advance\dimen@-\eqL@marginleft@
1655   \advance\dimen@-\eqL@shape@amount@
1656   \ifdim\eqL@fieldwidth@<\dimen@
1657     \dimen@\eqL@marginleft@
1658     \advance\dimen@\eqL@shape@amount@
1659     \eqL@adjust@print@alignleft\dimen@z@
1660   \else
1661     \eqL@adjust@print@alignrightz@z@
1662   \fi
1663 }

```

**TODO:** describe

```

1664 \def\eqL@adjust@center@tagsright@shoveleft@tag{%
1665   \dimen@\eqL@marginleft@
1666   \advance\dimen@\eqL@shape@amount@
1667   \advance\dimen@\eqL@tagwidth@
1668   \eqL@adjust@try\dimen@
1669   \ifnum\badness<\eqL@marginbadness@
1670     \dimen@\eqL@marginleft@
1671     \advance\dimen@\eqL@shape@amount@
1672     \eqL@adjust@print@alignleft\dimen@\eqL@tagwidth@
1673     \eqL@tagbox@print@right
1674   \else
1675     \ifdim\eqL@marginleft@>-\eqL@shape@amount@
1676       \eqL@adjust@try\eqL@tagwidth@
1677     \fi
1678     \ifnum\badness<\eqL@maxbadness@
1679       \eqL@adjust@print@alignrightz@\eqL@tagwidth@
1680       \eqL@tagbox@print@right
1681     \else
1682       \eqL@adjust@center@tagsright@shoveleft@notag
1683       \eqL@tagbox@print@right@raise
1684     \fi
1685   \fi
1686 }

```

**TODO:** describe

```

1687 \def\eqL@adjust@center@tagsleft@shoveright@notag{%
1688   \dimen@displaywidth
1689   \advance\dimen@-\eqL@tagmargin@
1690   \advance\dimen@-\eqL@marginright@
1691   \ifdim\eqL@fieldwidth@<\dimen@

```



```

1692   \eql@adjust@print@alignright\z@\eql@marginright@
1693   \else
1694   \eql@adjust@print@alignleft\z@\z@
1695   \fi
1696 }

```

**TODO:** describe

```

1697 \def\eql@adjust@center@tagsleft@shoveright@tag{%
1698   \dimen@\eql@marginright@
1699   \advance\dimen@\eql@tagwidth@
1700   \eql@adjust@try\dimen@
1701   \ifnum\badness<\eql@marginbadness@
1702     \eql@tagbox@print@left
1703     \eql@adjust@print@alignright\eql@tagwidth@\eql@marginright@
1704   \else
1705     \ifdim\eql@marginright@>\z@
1706       \eql@adjust@try\eql@tagwidth@
1707     \fi
1708     \ifnum\badness<\eql@maxbadness@
1709       \eql@tagbox@print@left
1710       \eql@adjust@print@alignleft\eql@tagwidth@\z@
1711     \else
1712       \eql@tagbox@print@left@raise
1713       \eql@adjust@center@tagsleft@shoveright@notag
1714     \fi
1715   \fi
1716   \eql@display@firstavail@set\z@
1717 }

```

**TODO:** describe

```

1718 \def\eql@adjust@center@tagsright@shoveright@notag{%
1719   \dimen@\displaywidth
1720   \advance\dimen@-\eql@tagmargin@
1721   \advance\dimen@-\eql@marginright@
1722   \ifdim\eql@fieldwidth@<\dimen@
1723     \dimen@\eql@tagmargin@
1724     \advance\dimen@\eql@marginright@
1725     \eql@adjust@print@alignright\z@\dimen@
1726   \else
1727     \eql@adjust@print@alignleft\z@\z@
1728   \fi
1729 }

```

**TODO:** describe

```

1730 \def\eql@adjust@center@tagsright@shoveright@tag{%
1731   \dimen@\eql@tagmargin@
1732   \advance\dimen@\eql@marginright@
1733   \ifdim\eql@tagwidth@<\dimen@
1734     \eql@adjust@try\dimen@%
1735     \ifnum\badness<\eql@marginbadness@
1736       \eql@adjust@print@alignright\z@\dimen@
1737       \eql@tagbox@print@right
1738     \else
1739       \eql@adjust@try\eql@tagwidth@
1740       \ifnum\badness<\eql@maxbadness@
1741         \eql@adjust@print@alignleft\z@\eql@tagwidth@
1742         \eql@tagbox@print@right
1743       \else

```

```

1744     \eql@adjust@print@alignleft\z@\z@
1745     \eql@tagbox@print@left@raise
1746   \fi
1747 \fi
1748 \else
1749   \eql@adjust@try\eql@tagwidth@
1750   \ifnum\badness<\eql@maxbadness@
1751     \eql@adjust@print@alignright\z@\eql@tagwidth@
1752     \eql@tagbox@print@right
1753   \else
1754     \eql@adjust@center@tagsright@shoveright@notag
1755     \eql@tagbox@print@right@raise
1756   \fi
1757 \fi
1758 }

```

**TODO:** describe

```

1759 \def\eql@adjust@center@tagsleft@shoveleft@notag{%
1760   \dimen@\displaywidth
1761   \advance\dimen@-\eql@tagmargin@
1762   \advance\dimen@-\eql@marginleft@
1763   \advance\dimen@-\eql@shape@amount@
1764   \ifdim\eql@fieldwidth@<\dimen@
1765     \dimen@\eql@tagmargin@
1766     \advance\dimen@\eql@marginleft@
1767     \advance\dimen@\eql@shape@amount@
1768     \eql@adjust@print@alignleft\dimen@\z@
1769   \else
1770     \eql@adjust@print@alignright\z@\z@
1771   \fi
1772 }

```

**TODO:** describe

```

1773 \def\eql@adjust@center@tagsleft@shoveleft@tag{%
1774   \dimen@\eql@tagmargin@
1775   \advance\dimen@\eql@marginleft@
1776   \advance\dimen@\eql@shape@amount@
1777   \ifdim\eql@tagwidth@<\dimen@
1778     \eql@adjust@try\dimen@%
1779     \ifnum\badness<\eql@marginbadness@
1780       \eql@tagbox@print@left
1781       \eql@adjust@print@alignleft\dimen@\z@
1782     \else
1783       \eql@adjust@try\eql@tagwidth@
1784       \ifnum\badness<\eql@maxbadness@
1785         \eql@tagbox@print@left
1786         \eql@adjust@print@alignright\eql@tagwidth@\z@
1787       \else
1788         \eql@tagbox@print@left@raise
1789         \eql@adjust@print@alignright\z@\z@
1790       \fi
1791     \fi
1792   \else
1793     \eql@adjust@try\eql@tagwidth@
1794     \ifnum\badness<\eql@maxbadness@
1795       \eql@tagbox@print@left
1796       \eql@adjust@print@alignleft\eql@tagwidth@\z@
1797     \else

```

```

1798     \eql@tagbox@print@left@raise
1799     \eql@adjust@center@tagsleft@shoveleft@notag
1800   \fi
1801 \fi
1802 \eql@display@firstavail@set\z@
1803 }

```

## J.5 Flush-Left Layout

**TODO:** describe

```

1804 \def\eql@adjust@flushleft@shoveleft{%
1805   \eql@marginleft@\eql@flushleftmargin@
1806   \advance\eql@marginleft@\eql@shape@amount@
1807   \ifdim\eql@marginleft@<\eql@flushleftmarginmin@
1808     \eql@marginleft@\eql@flushleftmarginmin@
1809   \fi
1810   \ifdim\eql@marginleft@>\eql@flushleftmarginmax@
1811     \eql@marginleft@\eql@flushleftmarginmax@
1812   \fi
1813 }

```

**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)

```

1814 \def\eql@adjust@flushleft@shoveleft@notag{%
1815   \ifdim\eql@flushleftmarginmin@<\eql@marginleft@
1816     \eql@adjust@try\eql@marginleft@
1817     \ifnum\badness<\eql@marginbadness@
1818       \eql@adjust@print@alignleft\eql@marginleft@\z@
1819     \else
1820       \eql@adjust@print@alignleft\eql@flushleftmarginmin@\z@
1821     \fi
1822   \else
1823     \eql@adjust@print@alignleft\eql@marginleft@\z@
1824   \fi
1825 }

```

**TODO:** describe

```

1826 \def\eql@adjust@flushleft@tagsright@shoveleft@notag{%
1827   \eql@adjust@flushleft@shoveleft
1828   \eql@adjust@flushleft@shoveleft@notag
1829 }
1830 \let\eql@adjust@flushleft@tagsleft@shoveleft@notag
1831   \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)

```

1832 \def\eql@adjust@flushleft@tagsright@shoveleft@tag{%
1833   \eql@adjust@flushleft@shoveleft
1834   \dimen@\eql@marginleft@
1835   \advance\dimen@\eql@tagwidth@
1836   \eql@adjust@try\dimen@
1837   \ifnum\badness<\eql@marginbadness@
1838     \eql@adjust@print@alignleft\eql@marginleft@\eql@tagwidth@

```

```

1839   \eql@tagbox@print@right
1840 \else
1841   \ifdim\eql@flushleftmarginmin<\eql@marginleft@
1842     \dimen@\eql@flushleftmarginmin@
1843     \advance\dimen@\eql@tagwidth@
1844     \eql@adjust@try\dimen@
1845   \fi
1846   \ifnum\badness<\eql@maxbadness@
1847     \eql@adjust@print@alignleft\eql@flushleftmarginmin@\eql@tagwidth@
1848     \eql@tagbox@print@right
1849   \else
1850     \eql@adjust@flushleft@shoveleft@notag
1851     \eql@tagbox@print@right@raise
1852   \fi
1853 \fi
1854 }

1855 \def\eql@adjust@flushleft@tagsleft@shoveleft@tag{%
1856   \eql@adjust@flushleft@shoveleft
1857   \ifdim\eql@tagwidth<\eql@flushleftmarginmin@
1858     \eql@tagbox@print@left
1859     \eql@adjust@flushleft@notag
1860   \else
1861     \ifdim\eql@tagwidth<\eql@marginleft@
1862       \eql@adjust@try\eql@marginleft@
1863       \ifnum\badness<\eql@marginbadness@
1864         \eql@tagbox@print@left
1865         \eql@adjust@print@alignleft\eql@marginleft@z@
1866       \else
1867         \eql@adjust@try\eql@tagwidth@
1868         \ifnum\badness<\eql@maxbadness@
1869           \eql@tagbox@print@left
1870           \eql@adjust@print@alignleft\eql@tagwidth@z@
1871         \else
1872           \eql@tagbox@print@left@raise
1873           \eql@adjust@print@alignleft\eql@flushleftmarginmin@z@
1874         \fi
1875       \fi
1876     \else
1877       \ifdim\eql@tagwidth>\eql@flushleftmarginmax@
1878         \eql@tagbox@print@left@raise
1879         \eql@adjust@flushleft@shoveleft@notag
1880       \else
1881         \eql@adjust@try\eql@tagwidth@
1882         \ifnum\badness<\eql@maxbadness@
1883           \eql@tagbox@print@left
1884           \eql@adjust@print@alignleft\eql@tagwidth@z@
1885         \else
1886           \eql@tagbox@print@left@raise
1887           \eql@adjust@flushleft@shoveleft@notag
1888         \fi
1889       \fi
1890     \fi
1891   \fi
1892   \eql@display@firstavail@setz@
1893 }

```

**TODO:** describe

```

1894 \def\eq@adjust@flushleft@shoveright@notag{%
1895   \eq@marginleft@\eq@flushleftmargin@
1896   \ifdim\eq@flushleftmarginmin<\eq@marginleft@
1897     \eq@adjust@try\eq@marginleft@
1898     \ifnum\badness<\eq@marginbadness@
1899       \eq@adjust@print@alignright\eq@marginleft@%z@
1900     \else
1901       \eq@adjust@print@alignright\eq@flushleftmarginmin@%z@
1902     \fi
1903   \else
1904     \eq@adjust@print@alignright\eq@marginleft@%z@
1905   \fi
1906 }
1907 \let\eq@adjust@flushleft@tagsright@shoveright@notag
1908   \eq@adjust@flushleft@shoveright@notag
1909 \let\eq@adjust@flushleft@tagsleft@shoveright@notag
1910   \eq@adjust@flushleft@shoveright@notag

```

**TODO:** describe

```

1911 \def\eq@adjust@flushleft@tagsright@shoveright@tag{%
1912   \dimen@\eq@marginleft@
1913   \advance\dimen@\eq@tagwidth@
1914   \eq@adjust@try\dimen@
1915   \ifnum\badness<\eq@marginbadness@
1916     \eq@adjust@print@alignright\eq@marginleft@\eq@tagwidth@
1917     \eq@tagbox@print@right
1918   \else
1919     \ifdim\eq@flushleftmarginmin<\eq@marginleft@
1920       \dimen@\eq@flushleftmarginmin@
1921       \advance\dimen@\eq@tagwidth@
1922       \eq@adjust@try\dimen@
1923     \fi
1924     \ifnum\badness<\eq@maxbadness@
1925       \eq@adjust@print@alignright\eq@flushleftmarginmin@\eq@tagwidth@
1926       \eq@tagbox@print@right
1927     \else
1928       \eq@adjust@flushleft@shoveright@notag
1929       \eq@tagbox@print@right@raise
1930     \fi
1931   \fi
1932 }

```

**TODO:** describe

```

1933 \def\eq@adjust@flushleft@tagsleft@shoveright@tag{%
1934   \ifdim\eq@tagwidth<\eq@flushleftmarginmin@
1935     \eq@tagbox@print@left
1936     \eq@adjust@flushleft@shoveright@notag
1937   \else
1938     \ifdim\eq@tagwidth<\eq@marginleft@
1939       \eq@adjust@try\eq@marginleft@
1940       \ifnum\badness<\eq@marginbadness@
1941         \eq@tagbox@print@left
1942         \eq@adjust@print@alignright\eq@marginleft@%z@
1943       \else
1944         \eq@adjust@try\eq@tagwidth@
1945         \ifnum\badness<\eq@maxbadness@
1946           \eq@tagbox@print@left
1947         \eq@adjust@print@alignright\eq@tagwidth@%z@

```

```

1948     \else
1949         \eql@tagbox@print@left@raise
1950         \eql@adjust@print@alignright\eql@flushleftmarginmin@z@
1951     \fi
1952 \fi
1953 \else
1954     \ifdim\eql@tagwidth@>\eql@flushleftmarginmax@
1955         \eql@tagbox@print@left@raise
1956         \eql@adjust@flushleft@shoveright@notag
1957     \else
1958         \eql@adjust@try\eql@tagwidth@
1959         \ifnum\badness<\eql@maxbadness@
1960             \eql@tagbox@print@left
1961             \eql@adjust@print@alignright\eql@tagwidth@z@
1962         \else
1963             \eql@tagbox@print@left@raise
1964             \eql@adjust@flushleft@shoveright@notag
1965         \fi
1966     \fi
1967 \fi
1968 \fi
1969 \eql@display@firstavail@set\z@
1970 }

1971 \def\eql@adjust@flushleft@shovecenter{%
1972 \eql@error{shove center not implemented for left alignment}%
1973 }
1974 \let\eql@adjust@flushleft@tagsright@shovecenter@notag
1975 \eql@adjust@flushleft@shovecenter
1976 \let\eql@adjust@flushleft@tagsright@shovecenter@tag
1977 \eql@adjust@flushleft@shovecenter
1978 \let\eql@adjust@flushleft@tagsleft@shovecenter@notag
1979 \eql@adjust@flushleft@shovecenter
1980 \let\eql@adjust@flushleft@tagsleft@shovecenter@tag
1981 \eql@adjust@flushleft@shovecenter

```

## K Single-Line Equation

**TODO:** describe

### K.1 Environment

```

1982 \def\eql@single@cr{%
1983 \eql@error{Cannot use '\string\\' within display equation.
1984 Please switch to equations environment}%
1985 }

```

**TODO:** describe

```

1986 \def\eql@single@start{%
1987 \eql@numbering@eval@mode
1988 \let\eql@numbering@subeq@use\eql@false
1989 \eql@stack@save@single

```

**TODO:** make other display environments push these!?

```

1990 \eql@numbering@single@init
1991 \ifdefined\eql@single@native

```

```

1992 \let\eql@single@start@sel\eql@single@start@native
1993 \let\eql@single@end@sel\eql@single@end@native
1994 \let\raisetag\eql@raisetag@default
1995 \else
1996 \let\eql@single@start@sel\eql@single@start@adjust
1997 \let\eql@single@end@sel\eql@single@end@adjust
1998 \fi
1999 \ifdefined\eql@single@crerror\else
2000 \let\\eql@single@cr
2001 \fi
2002 \eql@single@start@sel
2003 }

2004 \def\eql@single@end{%
2005 \eql@punct@apply@block
2006 \eql@hook@eqout
2007 \eql@single@end@sel
2008 \eql@stack@restore
2009 }

```

## K.2 Native

```

2010 \def\eql@single@start@native{%
2011 % \mathopen{}%
2012 \eql@hook@eqin
2013 }%

```

**TODO:** describe

```

2014 \def\eql@single@end@native{%
2015 % \mathclose{}%
2016 \if@eqnsw
2017 \ifdefined\eql@tagsleft
2018 \leqno
2019 \else
2020 \eqno
2021 \fi
2022 \eql@compose@print
2023 \fi
2024 \ifnum\eql@displaybreak@open@=\@MM\else
2025 \postdisplaypenalty\eql@displaybreak@open@
2026 \fi
2027 }%

```

## K.3 Adjustment

```

2028 \def\eql@single@start@adjust{%
2029 \eql@totalrows@\@ne
2030 \eql@row@\z@
2031 \eql@display@init
2032 \let\shoveleft\eql@adjust@shoveleft
2033 \let\shovecenter\eql@adjust@shovecenter
2034 \let\shoveright\eql@adjust@shoveright
2035 \ifdefined\eql@flushleft\else
2036 \eql@marginleft@\z@
2037 \eql@marginright@\z@
2038 \fi
2039 \eql@adjust@init
2040 \eql@shape@eval
2041 \eql@row@\@ne

```

```

2042 \setbox\eql@fieldbox@\hbox\bgroup
2043 \eql@restore@hfuzz
2044 \eql@strut@field
2045 $\m@th\displaystyle%$
2046 \eql@hook@eqin
2047 }

2048 \def\eql@single@end@adjust{%
2049   $%$
2050   \hfil
2051   \kern\z@
2052 \egroup

2053 \eql@fieldwidth@\wd\eql@fieldbox@
2054 \eql@line@height@\ht\eql@fieldbox@
2055 \eql@line@depth@\dp\eql@fieldbox@

2056 \eql@halign@init{}%
2057 \halign{##\cr
2058   \noalign{\eql@halign@before}%
2059   \if@eqnsw
2060     \eql@tagbox@make\eql@compose@print
2061     \eql@adjust@sel@tag
2062   \else
2063     \eql@adjust@sel@notag
2064   \fi
2065   \cr
2066   \noalign{\eql@halign@after}%
2067 }%
2068 \eql@row@\tw@
2069 \eql@display@close
2070 }

```

## L Multi-Line Support

**TODO:** describe

### L.1 Registers

```

\eql@column@
\eql@totalcolumns@
2071 \newcount\eql@column@
2072 \newcount\eql@totalcolumns@

\eql@totalwidth@ (dimen)
2073 \newdimen\eql@totalwidth@

\eql@line@width@ (dimen)
\eql@line@avail@ (dimen)
\eql@line@pos@ (dimen)
2074 \newdimen\eql@line@width@
2075 \newdimen\eql@line@avail@
2076 \newdimen\eql@line@pos@

\eql@fieldlength@tab
\eql@fieldlength@save
\eql@fieldlength@get
2077 \let\eql@fieldlength@tab\@empty
2078 \def\eql@fieldlength@save#1{%

```



```

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

```

```

\eq@tagwidth@get
\eq@tagwidth@save

```

```

2095 \let\eq@tagwidth@tab\@empty
2096 \def\eq@tagwidth@get#1{%
2097 \ifcase\expandafter#1\eq@tagwidth@tab\fi
2098 }
2099 \def\eq@tagwidth@save{%
2100 \begingroup
2101 \let\or\relax
2102 \global\edef\eq@tagwidth@tab{\eq@tagwidth@tab\or\the\eq@tagwidth@}%
2103 \endgroup
2104 }
2105 \def\eq@tagwidth@savezero{%
2106 \begingroup
2107 \let\or\relax
2108 \global\edef\eq@tagwidth@tab{\the\eq@tagwidth@\eq@tagwidth@tab}%
2109 \endgroup
2110 }

```

## L.2 Measure Support

**TODO:** describe

```

2111 \def\eq@measure@init#1{%
2112 \measuring@true
2113 \eq@row@\z@
2114 \let\displaybreak\eq@displaybreak@measure
2115 \tabskip\z@skip
2116 \everycr{%
2117 \noalign{%
2118 \global\advance\eq@row@\@ne
2119 #1%
2120 }%
2121 }%
2122 }

```

```

\eq@measure@restorecounters
\eq@measure@savecounters

```

```

2123 \let\eq@measure@restorecounters\@empty
2124 \def\eq@measure@savecounters{%
2125 \begingroup

```

```

2126 \def\@elt##1{%
2127   \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2128 \global\edef\@gtempa{%
2129   \cl@ckpt
2130   \let\noexpand\eq@measure@restorecounters\noexpand\@empty
2131 }%
2132 \endgroup
2133 \let\eq@measure@restorecounters\@gtempa
2134 }

```

### L.3 Print Support

**TODO:** describe

eq@print@inithalign

```

2135 \def\eq@print@init#1{%
2136   \eq@row@z@
2137   \eq@halign@init{%
2138     \global\eq@displaybreak@open@\@MM
2139     \global\advance\eq@row@\@ne
2140     #1%
2141   }%
2142 }

2143 \def\eq@print@overfull{%
2144   \dimen@eq@line@width@
2145   \advance\dimen@-\hfuzz
2146   \ifdim\dimen@>\displaywidth
2147     \setboxz@\hbox to\displaywidth{\hbox to\eq@line@width@{\hfil}}%
2148     \wdz@z@
2149     \htz@\eq@line@height@
2150     \dpz@\eq@line@depth@
2151     \boxz@
2152   \fi
2153 }

```

l@tagbox@print@multi

```

2154 \def\eq@tagbox@print@multi{%
2155   \advance\eq@tagwidth@-\eq@tagfuzz@
2156   \ifdefined\eq@tagsleft
2157     \eq@display@firstavail@setz@
2158     \ifdim\eq@tagwidth@>\eq@line@avail@
2159       \eq@tagbox@print@left@raise
2160     \else
2161       \eq@tagbox@print@left
2162     \fi
2163     \kern\displaywidth
2164   \else
2165     \kern\displaywidth
2166     \advance\eq@tagwidth@\eq@line@width@
2167     \ifdim\eq@tagwidth@>\displaywidth
2168       \eq@tagbox@print@right@raise
2169     \else
2170       \eq@tagbox@print@right
2171     \fi
2172   \fi
2173 }

```

## L.4 Line Breaks

**TODO:** describe

```
\eql@math@cr
2174 \protected\def\eql@math@cr{%
2175   \eql@ampprotecttwo\eql@teststaropt@tight
2176   {\global\eql@displaybreak@pen@ \@M\eql@math@cr}\eql@math@cr@z@}

\eql@math@cr@
2177 \def\eql@math@cr@[#1]{%
2178   \eql@math@cr@@@
2179   \cr

2180   \noalign{%
2181     \ifnum\eql@displaybreak@pen@=\@MM
2182       \penalty\interdisplaylinepenalty
2183     \else
2184       \penalty\eql@displaybreak@pen@
2185     \fi
2186     \addtolength\eql@vspaceskip@{#1}%
2187     \vskip\eql@vspaceskip@
2188     \global\eql@vspaceskip@z@skip
2189   }%
2190 }

\eql@let@cr
2191 \def\eql@let@cr#1{%
2192   \let\\eql@math@cr
2193   \let\eql@math@cr@@@#1%
2194 }
```

## L.5 Intertext

**TODO:** describe

```
2195 \eql@amsmath@futurebefore{\let\intertext\@undefined}
```

**TODO:** revert in everymath?

```
\intertext
\eql@intertext@
2196 \providecommand{\intertext}{\eql@error{Invalid use of \string\intertext}}

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

2197 \def\eql@intertext@process#1{%
2198   \eql@math@cr@@@
2199   \cr
2200   \ifmeasuring@\else
2201     \expandafter\eql@intertext@print{#1}%
2202   \fi
2203 }
```

**TODO:** describe

```
2204 \def\eql@intertext@print#1{%
2205   \noalign{%
2206     \eql@halign@after
2207     \eql@setkeys{intertext}\eql@intertext@opt
2208     \normalbaselines
2209     \penalty\postdisplaypenalty
2210     \addtolength\eql@vspaceskip@\belowdisplayskip
2211     \vskip\eql@vspaceskip@
2212     \global\eql@vspaceskip@\z@skip
2213     \vbox{%
2214       \@parboxrestore
2215       \ifdim
2216         \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2217       \else
2218         \parshape\@ne
2219         \@totalleftmargin\linewidth
2220       \fi
2221       \noindent
2222       \ignorespaces
2223       #1%
2224       \par
2225     }%
2226     \penalty\predisplaypenalty
2227     \vskip\abovedisplayskip
2228     \eql@halign@prevdepth@\z@
2229     \eql@halign@before
2230   }
2231 }
```

**TODO:** describe

```
2232 \newenvironment{eql@intertext}{%
2233   \eql@testopt@tight\eql@intertext@{}}%
2234 }{%
2235   \aftergroup\eql@intertext@after
2236   \ignorespacesafterend
2237 }
```

**TODO:** describe

```
2238 \def\eql@intertext@env{intertext}
2239 \def\eql@intertext@[#1]{%
2240   \global\def\eql@intertext@opt{#1}%
2241   \ifx\@currentvir\eql@intertext@env
2242     \expandafter\eql@scan@env\expandafter\eql@intertext@inject
2243   \else
2244     \expandafter\eql@intertext@process
2245   \fi
2246 }
```

**TODO:** describe

```
2247 \def\eql@intertext@inject{%
2248   \global\edef\eql@intertext@after{%
2249     \noexpand\eql@intertext@process{%
2250       \ifx\eql@scan@body\eql@scan@body@dump
2251         \eql@scan@body@dump
2252       \else
2253         \noexpand\scantokens{\eql@scan@body@dump}%

```

```

2254     \fi
2255   }%
2256 }%
2257 }

```

## L.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
```

```

2258 \let\eql@multi@mode@lines\eql@false
2259 \def\eql@multi@main{%
2260   \eql@numbering@eval@mode
2261   \eql@stack@save@multi
2262   \ifdefined\eql@subequations@active
2263     \let\eql@numbering@subeq@use\eql@false
2264   \fi
2265   \ifdefined\eql@numbering@subeq@use
2266     \eql@numbering@subeq@init
2267   \fi
2268   \let\intertext\eql@intertext
2269   \let\endintertext\endeql@intertext
2270   \let\shoveleft\eql@adjust@shoveleft
2271   \let\shovecenter\eql@adjust@shovecenter
2272   \let\shoveright\eql@adjust@shoveright
2273   \ifdefined\eql@multi@mode@lines
2274     \expandafter\eql@lines@measure
2275   \else
2276     \ifdefined\eql@ampproof@active
2277       \eql@ampproof
2278     \fi
2279     \expandafter\eql@align@measure
2280   \fi
2281   \ifx\eql@numbering@subeq@use\@ne
2282     \eql@numbering@subeq@revert
2283   \fi
2284   \ifdefined\eql@multi@mode@lines\else
2285     \ifdefined\eql@multi@linesfallback
2286       \ifnum\eql@totalcolumns@=\@ne
2287         \let\eql@multi@mode@lines\eql@true
2288         \eql@shape@set{r}%
2289       \eql@lines@measure
2290     \fi
2291   \fi
2292 \fi
2293 \ifdefined\eql@multi@mode@lines
2294   \expandafter\eql@lines@print
2295 \else
2296   \expandafter\eql@align@print
2297 \fi
2298 \ifdefined\eql@numbering@subeq@use
2299   \eql@numbering@subeq@close
2300 \fi
2301 \eql@stack@restore
2302 }

```

**TODO:** describe

```
2303 \def\eql@mode@equation{%
2304   \let\eql@equations@mode@single\eql@true
2305   \let\eql@equations@end\eql@single@end
2306 }
2307 \def\eql@mode@align{%
2308   \let\eql@equations@mode@single\eql@false
2309   \let\eql@multi@mode@lines\eql@false
2310   \let\eql@equations@end\@empty
2311 }
2312 \def\eql@mode@lines{%
2313   \let\eql@equations@mode@single\eql@false
2314   \let\eql@multi@mode@lines\eql@true
2315   \let\eql@equations@end\@empty
2316 }
2317 \eql@mode@align
```

## M Multi-Line Lines Mode

### M.1 Measure

**TODO:** describe

```
2318 \def\eql@lines@measure@line@begin{%
2319 (dev)\eql@dev{starting line \the\eql@row}%
2320   \eql@numbering@measure@line@begin
2321   \eql@hook@linein
2322 }
```

**TODO:** describe

```
2323 \def\eql@lines@measure@line@end{%
2324   \eql@punct@apply@line
2325   \eql@hook@lineout
2326 }
```

**TODO:** describe

```
2327 \def\eql@lines@measure@field{%
2328   \kern\wd\eql@fieldbox@
2329 }
```

**TODO:** describe

```
2330 \def\eql@lines@measure@tag{%
2331   \ifnum\eql@numbering@target@<\z@
2332     \if@eqnsw
2333       \eql@compose@null
2334     \fi
2335   \fi
2336 }
```

\eql@lines@measure

```
2337 \def\eql@lines@measure{%
2338 (dev)\eql@dev@center\eql@lines@measure
2339   \eql@measure@savecounters
2340   \setbox\z@\vbox{%
```

```

2341 \eql@numbering@measure@init
2342 \eql@measure@init\eql@lines@measure@line@begin
2343 \eql@let@cr\eql@lines@measure@line@end
2344 \halign{%
2345   \setbox\eql@fieldbox@\hbox{%
2346     \@lign
2347     $\m@th\displaystyle
2348     \eql@hook@colin
2349     ##%
2350     \eql@punct@apply@col
2351     \eql@hook@colout
2352     $%
2353   }%
2354 \eql@lines@measure@field
2355 \eql@lines@measure@tag
2356 \crr
2357 \noalign{%
2358   \eql@hook@blockbefore
2359 }%
2360 \eql@hook@blockin
2361 \eql@scan@body
2362 \ifvmode\else
2363   \eql@punct@apply@block
2364   \eql@hook@blockout
2365   \eql@lines@measure@line@end
2366 \cr
2367 \fi
2368 \omit
2369 \cr
2370 \noalign{%
2371   \eql@hook@blockafter
2372 }%
2373 }%
2374 \global\advance\eql@row@-\tw@
2375 \eql@numbering@measure@eval
2376 }%
2377 \eql@totalrows@\eql@row@
2378 \setbox\z@\vbox{%
2379   \unvbox\z@
2380   \unpenalty
2381   \global\setbox\@ne\lastbox
2382 }%
2383 \eql@totalwidth@\wd\@ne
2384 \ifdefined\eql@numbering@subeq@use
2385   \eql@numbering@subeq@test
2386 \fi
2387 \eql@measure@restorecounters
2388 <dev>\eql@dev@leave\eql@lines@measure
2389 }

```

## M.2 Print

**TODO:** describe

```
\eql@lines@math@cr
```

```

2390 \def\eql@lines@print@line@begin{%
2391 <dev>\eql@dev{starting line \the\eql@row@}%

```

```

2392 \eql@numbering@print@line@begin
2393 \eql@hook@linein
2394 }

```

**TODO:** describe

```

2395 \def\eql@lines@print@line@end{%
2396 \eql@punct@apply@line
2397 \eql@hook@lineout
2398 }

```

**TODO:** describe

```

2399 \def\eql@lines@print@line@adjust{%
2400 \eql@numbering@print@line@eval
2401 \eql@fieldwidth@\wd\eql@fieldbox@
2402 \eql@line@height@\ht\eql@fieldbox@
2403 \eql@line@depth@\dp\eql@fieldbox@
2404 \if@eqnsw
2405 \eql@tagbox@make\eql@compose@print
2406 \eql@adjust@sel@tag
2407 \else
2408 \eql@adjust@sel@notag
2409 \fi
2410 }

```

**TODO:** describe

```

2411 \def\eql@lines@print{%
2412 (dev)\eql@dev@center\eql@lines@print
2413 \eql@display@init
2414 \ifdefined\eql@flushleft
2415 % \TODO any init needed here? marginleft is used per line!
2416 \else
2417 \ifdefined\eql@paddingmax
2418 \eql@marginleft@z@
2419 \eql@marginright@z@
2420 \else
2421 \dimen@\displaywidth
2422 \advance\dimen@-\eql@totalwidth@
2423 \advance\dimen@-\eql@tagmargin@
2424 \divide\dimen@\tw@
2425 \eql@marginleft@\dimen@
2426 \advance\eql@marginleft@-\eql@paddingleft@
2427 \ifdim\eql@marginleft@<\z@
2428 \eql@marginleft@z@
2429 \fi
2430 \eql@marginright@\dimen@
2431 \advance\eql@marginright@-\eql@paddingright@
2432 \ifdim\eql@marginright@<\z@
2433 \eql@marginright@z@
2434 \fi
2435 \fi
2436 \fi
2437 \eql@adjust@init
2438 \eql@numbering@print@init
2439 \eql@print@init\eql@lines@print@line@begin
2440 \eql@let@cr\eql@lines@print@line@end
2441 \tabskipz@skip
2442 \halign{%
2443 \eql@shape@eval

```



```

2444     \setbox\eql@fieldbox@\hbox{%
2445     \eql@restore@hfuzz
2446     \eql@strut@ffield
2447     \@lign
2448     $\m@th\displaystyle
2449     \eql@hook@colin
2450     ##%
2451     \eql@punct@apply@col
2452     \eql@hook@colout
2453     $%
2454     \hfil
2455     \kern\z@
2456     }%
2457     \eql@lines@print@line@adjust
2458     \crr
2459     \noalign{%
2460     \eql@halign@before
2461     \eql@numbering@print@block@begin
2462     \eql@hook@blockbefore
2463     }%
2464 % \TODO relax? leavevmode?!
2465     \eql@hook@blockin
2466     \eql@scan@body
2467     \ifvmode\else
2468     \eql@punct@apply@block
2469     \eql@hook@blockout
2470     \eql@lines@print@line@end
2471     \cr
2472     \fi
2473     \noalign{%
2474     \eql@hook@blockafter
2475     \eql@halign@after
2476 (dev)\eql@dev@leave\eql@lines@print
2477     }%
2478     }%
2479     \eql@display@close
2480 }

```

## N Multi-Line Align Mode

**TODO:** describe

### N.1 Registers

**TODO:** describe

```
2481 \let\eql@align@margins\eql@true
```

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

```
2482 \newcount\eql@align@inter@
```

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

```
2483 \let\eql@colwidth@tab\@empty
```

`\eql@align@colwidth@get`

```
2484 \def\eql@align@colwidth@get#1{%
2485   \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2486 }
2487 \def\eql@align@colwidth@save{%
2488   \begingroup
2489     \let\or\relax
2490     \global\edef\eql@colwidth@tab{\or\the\wd\thr@\eql@colwidth@tab}%
2491   \endgroup
2492 }
```

## N.2 General Processing, Preamble

**TODO:** describe

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

```
2493 \def\eql@align@add@amp#1{\if m#1&\omit\expandafter\eql@align@add@amp\fi}
2494 \def\eql@align@completerow#1{%
2495   \begingroup
2496   \count@#1%
2497   \advance\count@-\eql@column@
2498   \advance\count@\@ne
2499   \edef\eql@tmp{\endgroup
2500     \expandafter\eql@align@add@amp\romannumeral\number\count@ 000q}%
2501   \eql@tmp
2502 }
```

`\eql@align@preamble`

```
2503 \def\eql@align@preamble#1#2#3{%
2504   &%
2505   \global\advance\eql@column@\@ne
2506   \hfil
2507   \global\setbox\eql@fieldbox@\hbox{%
2508     #3%
2509     \@lign
2510     $\m@th\displaystyle
2511     \eql@hook@colin
2512     ##%
2513     \eql@class@innerleft
2514     \eql@hook@innerleft
2515     $%
2516   }%
2517   \global\eql@fieldwidth@\wd\eql@fieldbox@
2518   #2%
2519   \tabskip\z@skip
2520   &%
2521   \global\advance\eql@column@\@ne
2522   \setbox\eql@fieldbox@\hbox{%
2523     % \TODO temp
2524     \kern-\wd\eql@fieldbox@
2525     \box\eql@fieldbox@
2526     #3%
2527     \@lign
2528     $\m@th\displaystyle
2529     \eql@hook@innerright
```

```

2530     \eql@class@innerright@sel
2531     ##%
2532     \eql@punct@apply@col
2533     \eql@hook@colout
2534     $%
2535     }%
2536     #2%
2537     \hfil
2538     \tabskip#1%
2539 }

2540 \def\eql@align@print@trailright{%
2541     &\omit
2542     \global\advance\eql@column@\@ne
2543     \setbox\eql@fieldbox@\hbox{%
2544 \kern-\wd\eql@fieldbox@\box\eql@fieldbox@
2545     }%
2546     \eql@align@print@field
2547 }

```

### N.3 Print

**TODO:** describe

```

2548 \def\eql@align@print@field{%

```

determine available and used space

```

2549 \dimen@eql@align@colwidth@get\eql@column@\relax
2550 \ifdim\wd\eql@fieldbox@>\z@
2551     \ifdim\eql@line@width@=\z@
2552         \eql@line@avail@\eql@line@pos@
2553         \ifodd\eql@column@
2554             \advance\eql@line@avail@\dimen@
2555             \advance\eql@line@avail@-\wd\eql@fieldbox@
2556         \fi
2557         \global\eql@line@avail@\eql@line@avail@
2558     \fi
2559     \eql@line@width@\eql@line@pos@
2560     \ifodd\eql@column@
2561         \advance\eql@line@width@\dimen@
2562     \else
2563         \advance\eql@line@width@\wd\eql@fieldbox@
2564     \fi
2565     \global\eql@line@width@\eql@line@width@
2566 \fi
2567 \advance\eql@line@pos@\dimen@
2568 \ifodd\eql@column@\else
2569     \advance\eql@line@pos@\eql@colsep@
2570 \fi
2571 \global\eql@line@pos@\eql@line@pos@

```

update height and depth

```

2572 \ifdim\ht\eql@fieldbox@>\eql@line@height@
2573     \global\eql@line@height@\ht\eql@fieldbox@
2574 \fi
2575 \ifdim\dp\eql@fieldbox@>\eql@line@depth@
2576     \global\eql@line@depth@\dp\eql@fieldbox@

```

2577 \fi

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

```
2578 %
2579 % \box\eq\fieldbox@
2580 %
2581 % \dimen@\eq\align@colwidth@get\eq\column@\relax
2582 % \advance\dimen@-\wd\eq\fieldbox@
2583 % \ifodd\eq\column@
2584 % \kern\dimen@
2585 % \box\eq\fieldbox@
2586 % \else
2587 % \box\eq\fieldbox@
2588 % \kern\dimen@
2589 % \fi
2590 %
2591 \dimen@\eq\align@colwidth@get\eq\column@\relax
2592 \ifodd\eq\column@
2593 \kern\dimen@
2594 \else
2595 \advance\dimen@-\wd\eq\fieldbox@
2596 \box\eq\fieldbox@
2597 \kern\dimen@
2598 \fi
2599 %
2600 }
```

ign@print@line@begin

```
2601 \def\eq\align@print@line@begin{%
2602 (dev)\eq\dev{starting line \the\eq\row@}%
2603 \global\eq\column@\z@
2604 \global\eq\line@pos@\eq\marginleft@
2605 \global\eq\line@width@\z@
2606 \global\eq\line@avail@\eq\totalwidth@
2607 \global\eq\line@height@\z@
2608 \global\eq\line@depth@\z@
2609 \eq\numbering@print@line@begin
2610 \eq\hook@linein
2611 }
```

align@print@line@end

```
2612 \def\eq\align@print@line@end{%
2613 \eq\punct@apply@line
2614 \eq\hook@lineout
2615 % \TODO add an even column with empty stuff if box processing deferred
2616 \ifodd\eq\column@
2617 \expandafter\eq\align@print@trailright
2618 \fi
2619 \eq\align@completerow\eq\totalcolumns@
2620 \eq\align@print@line
2621 }
```

eq\align@print@line

```
2622 \def\eq\align@print@line{%
```

```

2623 \dimen@ \eql@totalwidth@
2624 \advance\dimen@ \eql@colsep@
2625 \kern-\dimen@

determine first line available space

2626 \eql@display@firstavail@set\eql@line@avail@
2627 \eql@print@overflow@
2628 \eql@numbering@print@line@eval
2629 \if@eqnsw
2630 \eql@tagbox@make\eql@compose@print
2631 \eql@tagbox@print@multi
2632 \else
2633 \kern\displaywidth
2634 \fi
2635 }

```

`\eql@align@print`

```

2636 \def\eql@align@print{%
2637 (dev)\eql@dev@enter\eql@align@print
2638 \eql@align@adjust
2639 \eql@display@init
2640 \eql@numbering@print@init
2641 \eql@print@init\eql@align@print@line@begin
2642 \eql@let@cr\eql@align@print@line@end
2643 \tabskip\eql@marginleft@
2644 \halign{%
2645 \span\eql@align@preamble\eql@colsep@\eql@align@print@field\eql@strut@field
2646 \cr
2647 \noalign{%
2648 \eql@halign@before
2649 \eql@numbering@print@block@begin
2650 \eql@hook@blockbefore
2651 }%
2652 \eql@hook@blockin
2653 \eql@scan@body
2654 \ifvmode\else
2655 \eql@punct@apply@block
2656 \eql@hook@blockout
2657 \eql@align@print@line@end
2658 \cr
2659 \fi
2660 \noalign{%
2661 \eql@hook@blockafter
2662 \eql@halign@after
2663 (dev)\eql@dev@leave\eql@align@print
2664 }%
2665 }%
2666 \eql@display@close
2667 }

```

## N.4 Measure

**TODO:** describe

`n@measure@line@begin`

```

2668 \def\eql@align@measure@line@begin{%
2669 <dev>\eql@dev{starting line \the\eql@row}%
2670 \global\eql@column@z@
2671 \eql@numbering@measure@line@begin
2672 \eql@hook@linein
2673 }

```

```

2674 \def\eql@align@measure@field{%
2675 \eql@fieldlength@save\eql@column@
2676 \kern\wd\eql@fieldbox@
2677 }

```

ign@measure@line@end

```

2678 \def\eql@align@measure@line@end{%
2679 \eql@punct@apply@line
2680 \eql@hook@lineout
2681 &\omit
2682 \ifnum\eql@column@>\eql@totalcolumns@
2683 \global\eql@totalcolumns@\eql@ccolumn@
2684 \fi
2685 \eql@align@measure@tag
2686 }

```

ql@align@measure@tag

```

2687 \def\eql@align@measure@tag{%
2688 \ifnum\eql@numbering@target<\z@
2689 \if@eqnsw
2690 \eql@tagbox@make\eql@compose@measure
2691 \else
2692 \eql@tagwidth@z@
2693 \fi
2694 \eql@tagwidth@save
2695 \fi
2696 }

```

\eql@align@measure

```

2697 \def\eql@align@measure{%
2698 <dev>\eql@dev@enter\eql@align@measure
2699 \eql@totalcolumns@z@
2700 \let\eql@tagwidth@tab@empty
2701 \let\eql@fieldlength@tab@empty
2702 \eql@measure@savecounters
2703 \setbox\z@\vbox{%
2704 \eql@numbering@measure@init
2705 \eql@measure@init\eql@align@measure@line@begin
2706 \eql@let@cr\eql@align@measure@line@end
2707 \halign{%
2708 \span\eql@align@preamble\z@skip\eql@align@measure@field\relax
2709 \crr
2710 \noalign{%
2711 \eql@hook@blockbefore
2712 }%
2713 \eql@hook@blockin
2714 \eql@scan@body

```

**TODO:** test for vmode okay?!

```

2715     \ifvmode\else
2716         \eql@punct@apply@block
2717         \eql@hook@blockout
2718         \eql@align@measure@line@end
2719         \cr
2720     \fi
2721     \noalign{%
2722         \eql@hook@blockafter
2723     }%

```

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

```

2724     \eql@align@completerow\eql@totalcolumns@
2725     \cr
2726 }%
2727 \global\advance\eql@row@-\tw@
2728 \eql@numbering@measure@eval
2729 \ifnum\eql@numbering@target@>\z@
2730     \eql@tagbox@make\eql@compose@measure
2731     \eql@tagwidth@savezero
2732 \fi
2733 }%
2734 \eql@totalrows@\eql@row@
2735 \ifdefined\eql@numbering@subeq@use
2736     \eql@numbering@subeq@test
2737 \fi
2738 \eql@measure@restorecounters
2739 % \eql@totalwidth@\wd\z@

2740 \setbox\z@\vbox{%
2741     \unvbox\z@
2742     \unpenalty
2743     \global\setbox\@ne\lastbox
2744 }%

2745 % \TODO or this one?!
2746 \eql@totalwidth@\wd\@ne

```

**TODO:** why not recycle box contents altogether?!

```

2747 \let\eql@colwidth@tab\@empty
2748 \loop
2749     \setbox\@ne\hbox{%
2750         \unhbox\@ne
2751         \unskip
2752         \global\setbox\thr@\@lastbox
2753     }%
2754 \ifhbox\thr@@
2755     \eql@align@colwidth@save
2756 \repeat
2757 (dev)\eql@dev@leave\eql@align@measure
2758 }

```

## N.5 Adjust

**TODO:** describe

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

```
2759 \def\eq@align@adjust{%
```

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

```
2760 \ifodd\eq@totalcolumns@
2761 \advance\eq@totalcolumns@\@ne
2762 \fi
```

**TODO:** should we guard against no columns?!

```
2763 \ifnum\eq@totalcolumns@<\thr@@
2764 \let\eq@align@margin\eq@true
2765 \fi
```

```
2766 \eq@align@inter@\eq@totalcolumns@
2767 \divide\eq@align@inter@\tw@
2768 \advance\eq@align@inter@\m@ne
```

```
2769 \eq@colsep@\displaywidth
2770 \advance\eq@colsep@-\eq@totalwidth@
2771 \ifdefined\eq@flushleft
2772 \advance\eq@colsep@-\eq@flushleftmargin@
2773 \fi
2774 \count@\eq@align@inter@
2775 \ifdefined\eq@align@margin
2776 \ifdefined\eq@flushleft
2777 \advance\count@\@ne
2778 \else
2779 \advance\count@\tw@
2780 \fi
2781 \fi
2782 \divide\eq@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)

```
2783 \ifdim\eq@colsep@<\eq@colsepmin@
2784 \eq@colsep@\eq@colsepmin@
2785 \else
2786 \ifdim\eq@colsep@>\eq@colsepmax@
2787 \eq@colsep@\eq@colsepmax@
2788 \fi
2789 \fi

2790 \ifdefined\eq@flushleft
2791 \ifdim\eq@colsep@=\eq@colsepmin@
2792 \eq@marginleft@\displaywidth
2793 \advance\eq@marginleft@-\eq@totalwidth@
2794 \advance\eq@marginleft@-\eq@align@inter@\eq@colsep@
2795 \ifdim\eq@marginleft@>\eq@flushleftmargin@
2796 \eq@marginleft@\eq@flushleftmargin@
2797 \else
2798 \ifdim\eq@marginleft@<\eq@flushleftmarginmin@
2799 \eq@marginleft@\eq@flushleftmarginmin@
2800 \fi
2801 \fi
2802 \else
2803 \eq@marginleft@\eq@flushleftmargin@
2804 \fi
```



```

2805 \else
2806 \ifdefined\eq@align@argins
2807 \eq@marginleft@displaywidth
2808 \advance\eq@marginleft@-\eq@totalwidth@
2809 \advance\eq@marginleft@-\eq@align@inter@\eq@colsep@
2810 \ifdim\eq@marginleft@<\eq@tagmargin@
2811 \ifdim\ifdefined\eq@tagsleft\eq@marginleft@else-\p@fi<\z@
2812 \eq@marginleft@\z@
2813 \fi
2814 \else
2815 \advance\eq@marginleft@-\eq@tagmargin@
2816 \divide\eq@marginleft@tw@
2817 \fi
2818 \else
2819 \eq@marginleft@\z@
2820 \fi
2821 \fi

2822 \ifdefined\eq@tagsleft
2823 \let\eq@align@adjust@test\eq@align@adjust@test@tagleft
2824 \else
2825 \let\eq@align@adjust@test\eq@align@adjust@test@tagright
2826 \fi
2827 \loop\ifnum\eq@row@>\z@
2828 \ifnum\eq@numbering@target@<\z@
2829 \eq@tagwidth@\eq@tagwidth@get\eq@row@relax
2830 \else
2831 \ifnum\eq@numbering@target@=\eq@row@
2832 \eq@tagwidth@\eq@tagwidth@get\z@relax
2833 \fi
2834 \fi
2835 \ifdim\eq@tagwidth@>\z@
2836 \eq@align@adjust@width
2837 \eq@align@adjust@test
2838 \fi
2839 \advance\eq@row@m@ne
2840 \repeat

2841 \advance\eq@totalwidth@\eq@align@inter@\eq@colsep@
2842 \advance\eq@totalwidth@\eq@marginleft@
2843 }

```

**Calc Space. TODO:** describe

```

2844 \def\eq@align@adjust@width{%
2845 \eq@line@pos@\z@
2846 \eq@column@\z@
2847 \eq@line@avail@\eq@totalwidth@
2848 \@tempcnta\eq@totalcolumns@
2849 \eq@line@width@\z@
2850 \@tempcntb\z@
2851 \edef\@tempb{\eq@fieldlength@get\eq@row@}%
2852 \@for\@tempa:=\@tempb\do
2853 \eq@align@adjust@width@col
2854 \advance\@tempcnta@m@ne
2855 \divide\@tempcnta\tw@
2856 \advance\@tempcntb@m@ne
2857 \divide\@tempcntb\tw@
2858 }

```

**TODO:** describe

```
2859 \def\eql@align@adjust@width@col{%
2860   \advance\eql@column@\@ne
2861   \@tempdima\@tempa\relax
2862   \dimen@\eql@align@colwidth@get\eql@column@\relax
2863   \ifdim\@tempdima>\z@
2864     \ifdim\eql@line@width@=\z@
2865       \eql@line@avail@\eql@line@pos@
2866       \@tempcnta\eql@column@
2867       \ifodd\eql@column@
2868         \advance\eql@line@avail@\dimen@
2869         \advance\eql@line@avail@-\@tempdima
2870       \fi
2871     \fi
2872     \eql@line@width@\eql@line@pos@
2873     \@tempcntb\eql@column@
2874     \ifodd\eql@column@
2875       \advance\eql@line@width@\dimen@
2876     \else
2877       \advance\eql@line@width@\@tempdima
2878     \fi
2879   \fi
2880   \advance\eql@line@pos@\dimen@
2881 }
```

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

@align@adjust@test@r

```
2882 \def\eql@align@adjust@test@tagright{%
2883   \@tempdima\eql@line@width@
2884   \advance\@tempdima\eql@tagwidth@
2885   \dimen@\eql@marginleft@
2886   \advance\dimen@\@tempdima
2887   \advance\dimen@\@tempcntb\eql@colsep@
2888   \ifdim\dimen@>\displaywidth
2889     \eql@align@adjust@modify@tagright
2890   \fi
2891 }
```

**TODO:** describe

```
2892 \def\eql@align@adjust@modify@tagright{%
2893   \dimen@\eql@colsepmin@
2894   \multiply\dimen@\@tempcntb
2895   \advance\dimen@\@tempdima
2896   \ifdefined\eql@flushleft
2897     \advance\dimen@\eql@flushleftmarginmin@
2898   \fi
2899   \ifdim\dimen@>\displaywidth\else
```

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

```
2900   \ifnum\@tempcntb>\z@
2901     \dimen@\displaywidth
2902     \advance\dimen@-\@tempdima
```

```

2903     \count@\@tempcntb
2904     \ifdefined\eq\@flushleft
2905         \advance\dimen@-\eq\@marginleft@
2906     \else
2907         \ifdefined\eq\@align@margins
2908             \advance\count@\@ne
2909         \fi
2910     \fi
2911     \divide\dimen@\count@
2912     \ifdim\dimen@<\eq\@colsep@
2913         \ifdim\dimen@<\eq\@colsepmin@
2914             \dimen@\eq\@colsepmin@
2915         \fi
2916         \eq\@colsep@\dimen@
2917     \fi
2918 \fi

```

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

```

2919     \dimen@\displaywidth
2920     \advance\dimen@-\@tempdima
2921     \advance\dimen@-\@tempcntb\eq\@colsep@
2922     \ifdim\dimen@<\eq\@marginleft@
2923         \eq\@marginleft@\dimen@
2924     \fi
2925 \fi
2926 }

```

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

```

2927 \def\eq\@align@adjust@test@tagleft{%
2928     \count@\eq\@align@inter@
2929     \advance\count@-\@tempcnta
2930     \@tempdima-\eq\@line@avail@
2931     \advance\@tempdima\eq\@tagwidth@
2932     \dimen@\eq\@marginleft@
2933     \advance\dimen@\@tempcnta\eq\@colsep@
2934     \ifdim\dimen@<\@tempdima
2935         \eq\@align@adjust@modify@tagleft
2936     \fi
2937 }

```

```

2938 \def\eq\@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?!

```

2939     \ifdefined\eq\@flushleft
2940         \dimen@\eq\@flushleftmarginmax@
2941     \else
2942         \dimen@\displaywidth
2943     \fi
2944     \ifdim\dimen@>\eq\@tagwidth@
2945         \dimen@\eq\@colsepmin@
2946         \multiply\dimen@\count@
2947         \advance\dimen@\eq\@totalwidth@
2948         \advance\dimen@\@tempdima
2949         \ifdim\dimen@>\displaywidth\else

```

```

2950     \ifnum\count@>0
2951         \dimen@displaywidth
2952         \advance\dimen@-\eql@totalwidth@
2953         \advance\dimen@-\@tempdima
2954         \ifdefined\eql@align@argins
2955             \advance\count@\@ne
2956         \fi
2957         \divide\dimen@\count@
2958         \ifdim\dimen@<\eql@colsep@
2959             \ifdim\dimen@<\eql@colsepmin@
2960                 \dimen@\eql@colsepmin@
2961             \fi
2962             \advance\dimen@-\eql@colsep@
2963             \advance\eql@marginleft@-\eql@align@inter@\dimen@
2964             \advance\eql@colsep@\dimen@
2965         \fi
2966     \fi

2967     \dimen@-\@tempcnta\eql@colsep@
2968     \advance\dimen@\@tempdima
2969     \ifdim\dimen@>\eql@marginleft@
2970         \eql@marginleft@\dimen@
2971     \fi
2972 \fi
2973 \fi
2974 }

```

## O Interface

### O.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.

```
2975 \newtoks\eql@scan@reg@
```

**TODO:** describe

```

2976 \def\eql@scan@body@dump{\the\eql@scan@reg@}
2977 \def\eql@scan@body@rescan{%
2978   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
2979 \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`:

```
2980 \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:

```
2981 \def\eql@scan@env#1{%
```

```

2982 (dev)\eql@dev@enter\eql@scan@env
2983 \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenvir}}%
2984 \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`:

```

2985 % \begingroup
2986 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
2987 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
2988 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
2989 \eql@scan@iterate
2990 }

```

`\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.

```

2991 \def\eql@scan@env@count#1\begin#2{%
2992 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
2993 }

```

`\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:

```

2994 \def\eql@scan@env@iterate#1\end#2{%
2995 \edef\eql@scan@stack{%
2996 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
2997 \ifx\@empty\eql@scan@stack
2998 \@checkend{#2}%
2999 \eql@scan@addto{#1}%
3000 \expandafter\let\eql@scan@iterate\eql@scan@env@org
3001 (dev)\eql@dev@leave\eql@scan@env
3002 \expandafter\eql@scan@end
3003 \else
3004 \eql@scan@addto{#1\end{#2}}%
3005 \expandafter\eql@scan@iterate
3006 \fi
3007 }

```

**TODO:** describe

```

3008 \def\eql@scan@env@cancel{%
3009 \@namedef{end\@currenvir}{\ignorespacesafterend}%
3010 }

```

square brackets **TODO:** describe

```

3011 \def\eql@scan@sqr#1{%
3012 (dev)\eql@dev@enter\eql@scan@sqr
3013 \def\eql@scan@end{#1\]}%
3014 \eql@scan@reg@{\def\eql@scan@stack{b}}%

```

```

3015 \let\eql@scan@sqr@org\[%\]
3016 \let\[\eql@scan@sqr@iterate%\]
3017 \[%\]
3018 }

```

**TODO:** describe

```

3019 \def\eql@scan@sqr@count#1\[#2%\]
3020 \ifx\#2\else b\expandafter\eql@scan@sqr@count\fi
3021 }

```

**TODO:** describe

```

3022 \def\eql@scan@sqr@iterate#1\{%
3023 \edef\eql@scan@stack{%
3024 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
3025 \ifx\@empty\eql@scan@stack
3026 \let\[\eql@scan@sqr@org%\]
3027 \eql@scan@addto{#1}%
3028 (dev)\eql@dev@leave\eql@scan@sqr
3029 \expandafter\eql@scan@end
3030 \else
3031 \eql@scan@addto{#1\}%
3032 \expandafter\[%\]
3033 \fi
3034 }

3035 \def\eql@scan@sqrang@cancel{\ignorespaces}

```

**anuglar brackets** **TODO:** describe

```

3036 \def\eql@scan@ang#1{%
3037 (dev)\eql@dev@enter\eql@scan@ang
3038 \def\eql@scan@end{#1\>}%
3039 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
3040 \let\eql@scan@ang@org\<%\>
3041 \let\<\eql@scan@ang@iterate%\>
3042 \<%\>
3043 }

```

**TODO:** describe

```

3044 \def\eql@scan@ang@count#1\<#2%\>
3045 \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
3046 }

```

**TODO:** describe

```

3047 \def\eql@scan@ang@iterate#1\>{%
3048 \edef\eql@scan@stack{%
3049 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
3050 \ifx\@empty\eql@scan@stack
3051 \let\<\eql@scan@ang@org%\>
3052 \eql@scan@addto{#1}%
3053 (dev)\eql@dev@leave\eql@scan@ang
3054 \expandafter\eql@scan@end
3055 \else
3056 \eql@scan@addto{#1\>}%
3057 \expandafter\<%\>
3058 \fi
3059 }

```

## O.2 Options Processing

```
3060 \def\eq@equations@testall{\eq@equations@testtilde}
3061 \def\eq@equations@testtilde#1{%
3062   \eq@ifnextgobble@tight~%
3063   {\eqnaddopt{lines}\eq@equations@testopt{#1}}%
3064   {\eq@equations@testopt{#1}}}
3065 \def\eq@equations@testopt#1{%
3066   \eq@ifnextchar@tight[%
3067     {\eq@equations@addopt{\eq@equations@testexcl{#1}}}%
3068     {\eq@equations@testexcl{#1}}}
3069 \def\eq@equations@addopt#1[#2]{\eqnaddopt{#2}#1}
3070 \def\eq@equations@testexcl#1{%
3071   \eq@ifnextgobble@tight!%
3072   {\eqnaddopt{donumber}\eq@equations@testat{#1}}%
3073   {\eq@equations@teststar{#1}}}
3074 \def\eq@equations@teststar#1{%
3075   \eq@ifstar@tight%
3076   {\eqnaddopt{nonumber}\eq@equations@testat{#1}}%
3077   {\eq@equations@testat{#1}}}
3078 \def\eq@equations@testat#1{%
3079   \eq@ifat@tight
3080   {\eq@equations@addlabel{#1}}%
3081   {#1}}
3082 \def\eq@equations@addlabel#1#2{\eqnaddopt{label={#2}}#1}
3083 \def\eq@equations@processopt{%
3084   \let\eq@blocklabel\@undefined
3085   \let\eq@blocktag\@undefined
3086   \let\eq@display@doatend\@empty
3087   \eq@abovespace@\z@skip
3088   \eq@belowspace@\z@skip
3089   \eq@displaybreak@prepen@\@MM
3090   \eq@nextopt@process{equations}%
3091   \let\eq@punct@block\eq@punct@main
3092   \let\eq@punct@main\relax
3093   \setlength\eq@paddingleft@{\eq@paddingleft@val}
3094   \setlength\eq@paddingright@{\eq@paddingright@val}
3095   \setlength\eq@indent@{\eq@indent@val}
3096   \setlength\eq@colsepmin@{\eq@colsepmin@val}
3097   \setlength\eq@colsepmax@{\eq@colsepmax@val}
3098 }
```

## O.3 Environments

equations (*env.*)

```
3099 \newenvironment{equations}{%
3100   <dev>\eq@dev@enterenv
3101   \ifmmode
3102     \eq@error@mathmode{\string\begin{\@currenvir}}%
3103     \expandafter\eq@scan@env\expandafter\eq@scan@env@cancel
3104   \else
3105     \eq@halign@catchprevdepth
3106     $$$$$
3107     \expandafter\eq@ampprotect\expandafter\eq@equations@testall
3108     \expandafter\eq@equations@start
3109   \fi
3110 }{%
3111   \eq@equations@end
```

```

3112 $$$
3113 \ignorespacesafterend
3114 (dev)\eql@dev@leaveenv
3115 }

3116 \def\eql@equations@start{%
3117 \eql@equations@processopt
3118 \ifdefined\eql@equations@mode@single
3119 \expandafter\eql@single@start
3120 \else
3121 \expandafter\eql@scan@env\expandafter\eql@multi@main
3122 \fi
3123 }

```

### square and angle brackets

```

3124 \def\eql@equations@sqr@opt{equation,nonumber}
3125 \protected\def\eql@equations@sqr@open{%
3126 \ifmode
3127 \eql@error@mathmode{\string\[\dots\string\]}%
3128 \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
3129 \else
3130 (dev)\eql@dev@enter{\[\dots\string\]}%
3131 \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
3132 \eql@halign@catchprevdepth
3133 $$$
3134 \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3135 \expandafter\eql@equations@sqr@start
3136 \fi
3137 }

```

```

3138 \def\eql@equations@ang@opt{align,nonumber}
3139 \protected\def\eql@equations@ang@open{%
3140 \ifmode
3141 \eql@error@mathmode{\string\langle\dots\string\rangle}%
3142 \expandafter\eql@scan@ang\expandafter\eql@scan@sqrang@cancel
3143 \else
3144 (dev)\eql@dev@enter{\langle\dots\string\rangle}%
3145 \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
3146 \eql@halign@catchprevdepth
3147 $$$
3148 \expandafter\eql@ampprotect\expandafter\eql@equations@testall
3149 \expandafter\eql@equations@ang@start
3150 \fi
3151 }

```

```

3152 \def\eql@equations@sqr@start{%
3153 \eql@equations@processopt
3154 \ifdefined\eql@equations@mode@single
3155 \expandafter\eql@single@start
3156 \else
3157 \expandafter\eql@scan@sqr\expandafter\eql@multi@main
3158 \fi
3159 }

```

```

3160 \def\eql@equations@ang@start{%
3161 \eql@equations@processopt
3162 \ifdefined\eql@equations@mode@single
3163 \expandafter\eql@single@start

```



```

3164 \else
3165   \expandafter\eql@scan@ang\expandafter\eql@multi@main
3166 \fi
3167 }

3168 \protected\def\eql@equations@sqr@close{%
3169 \eql@equations@end
3170 $$$}
3171 (dev)\eql@dev@leave{\[...\string\]}%
3172 \ignorespaces
3173 }

3174 \protected\def\eql@equations@ang@close{%
3175 \eql@equations@end
3176 $$$}
3177 (dev)\eql@dev@leave{\<...\string\>}%
3178 \ignorespaces
3179 }

```

## O.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 (,)?

```

3180 \DeclareRobustCommand{\eql@eqref}[1]{\textup{\tagform@{\ref{#1}}}}

```

`\numberwithin` amsmath defines the macro `\numberwithin` to specify that equations should be numbered within sections. The L<sup>A</sup>T<sub>E</sub>X kernel contains a similar command `\counterwithin` (with a slightly extended syntax) that can be used as a drop-in replacement for `\numberwithin`:

```

3181 \DeclareRobustCommand{\eql@numberwithin}[3][\arabic]{%
3182 \@ifundefined{c@#2}{\@nocounterr{#2}}{%
3183 \@ifundefined{c@#3}{\@nocounterr{#3}}{%
3184 \ifdefined\HyOrg@addtoreset
3185 \HyOrg@addtoreset{#2}{#3}%
3186 \else
3187 \@addtoreset{#2}{#3}%
3188 \fi
3189 \expandafter\xdef\csname the#2\endcsname{%
3190 \expandafter\noexpand\csname the#3\endcsname.\noexpand#1{#2}}%
3191 \ifdefined\theHequation
3192 \expandafter\xdef\csname theH#2\endcsname{%
3193 \expandafter\noexpand
3194 \csname the\@ifundefined{theH#3}{}H#3\endcsname.%
3195 \noexpand\arabic{#2}}%
3196 \fi
3197 }%
3198 }%
3199 }

```

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

```

3200 \DeclareRobustCommand{\eql@allowdisplaybreaks}[1][4]{%
3201 \eqnlineset{allowbreaks=#1}%
3202 }

```

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

```
3203 \newcommand{\eql@notag}{\nonumber}
```

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

```
3204 \newcommand{\eql@thetag}{\leavevmode\tagform@}
```

## P Options

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

```
3205 \RequirePackage{keyval}
```

### P.1 Selection Tools

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

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

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

```
3206 \def\eql@decide@select#1#2#3#4{%
3207   \def\@tempa{#3}%
3208   \let\@tempd\@undefined
3209   \@for\@tempc:=#4\do{%
3210     \ifdefined\@tempd\else
3211       \edef\@tempb{\noexpand\@tempb:=\expandafter\@firstoftwo\@tempc}%
3212       \expandafter\@for\@tempb\do{%
3213         \ifx\@tempa\@tempb
3214           \expandafter\expandafter\expandafter\def
3215             \expandafter\expandafter\expandafter\@tempd
3216               \expandafter\expandafter\expandafter{%
3217                 \expandafter\@secondoftwo\@tempc}%
3218         \fi
3219       }%
3220     \fi
3221   }%
3222   \ifdefined\@tempd
3223     \@tempd
3224   \else
3225     \eql@error{undefined value ‘#3’ for option ‘#2’ of ‘#1’}%
3226   \fi
3227 }
```

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

```
3228 \def\eql@decide@if#1#2#3#4#5{%
3229   \eql@decide@select{#1}{#2}{#3}{%
3230     \{on,true,yes,enabled,1\}{#4}},%
3231     \{off,false,no,disabled,0\}{#5}}}
```

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

```
3232 \def\eql@decide@bool#1#2#3#4{%
3233   \eql@decide@if{#1}{#2}{#3}{\let#4\eql@true}{\let#4\eql@false}}
```

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

```
3234 \def\eql@decide@abovebelow#1#2#3#4#5{%
3235   \eql@decide@select{#1}{#2}{#3}{%
3236     {,abovebelow,both,tb}{#4#5},%
3237     {above,top,t}{#4},%
3238     {below,bottom,b}{#5}}%
3239 }
```

## P.2 Declaration Code

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

```
3240 \def\eql@define@key#1#2{%
3241   \eql@ifnextchar@loose[%]
3242     {\eql@definekey@opt{#1}{#2}}%
3243     {\eql@definekey@noopt{#1}{#2}}%
3244 }
3245 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{#3}}
3246 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{#3}{#4}}
3247 \def\eql@definekey@for#1#2#3{%
3248   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}{#3}}%
3249   \edef\eql@for@vara{\noexpand\eql@for@vara:=#1}%
3250   \expandafter\@for\eql@for@vara\do{%
3251     \edef\eql@for@varb{\noexpand\eql@for@varb:=#2}%
3252     \expandafter\@for\eql@for@varb\do{%
3253       \edef\eql@for@call##1{%
3254         \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@vara}}%
3255       \eql@for@call{##1}}%
3256     }%
3257   }%
3258 }
```

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

```
3259 \def\eql@setkeys#1#2{
3260   \def\eql@tmp{\setkeys{eql@#1}}%
3261   \expandafter\eql@tmp\expandafter{#2}%
3262 }
```

`\eql@nextopt` It can be convenient to add arguments to the following equations environment, e.g.

`\eql@nextopt@process` towards defining modifier macros:

`\eqnaddopt`

```
3263 \let\eql@nextopt\@empty
3264 \def\eql@nextopt@process#1{%
3265   (dev)\eql@dev@start\eql@nextopt@process
3266   \eql@setkeys{#1}\eql@nextopt
3267   \global\let\eql@nextopt\@empty
3268 }
3269 \newcommand{\eqnaddopt}[1]{%
3270   \expandafter\def\expandafter\eql@nextopt\expandafter{\eql@nextopt,#1}}
```

## P.3 Options Declarations

**TODO:** describe

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

```
3271 \eql@define@key{equationsbox}{gathered,gather,ga,lines,lined,ln,\string~}[]{%
3272   \eql@mode@lined}
3273 \eql@define@key{equationsbox}{aligned,align,al,columns,col,@}[]{%
3274   \eql@mode@aligned}
3275 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
3276 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
3277 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
3278 \eql@define@key{equationsbox}{colsep}{\def\eql@box@colsep{#1}}
```

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

```
3279 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@equation}
3280 \eql@define@key{equations}{gathered,gather,ga,lines,lined,ln,\string~}[]{%
3281   \eql@mode@lines}
3282 \eql@define@key{equations}{aligned,align,al,columns,col,@}[]{%
3283   \eql@mode@align}
3284 \eql@define@key{equations}{native}[true]{%
3285   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
3286   \ifdefined\eql@single@native\let\eql@flushleft\eql@false\fi}
3287 \eql@define@key{setup}{native}[true]{%
3288   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
3289 \eql@define@key{setup}{sqr@opt}[]{%
3290   \def\eql@equations@sqr@opt{equation,#1}}
3291 \eql@define@key{setup}{ang@opt}[]{%
3292   \def\eql@equations@ang@opt{align,#1}}
```

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

```
3293 \def\eql@keycat{equations,equationsbox,setup}
3294 \eql@define@key\eql@keycat{s@spread}[\jot]{\def\eql@s@spread{#1}}
3295 \eql@define@key\eql@keycat{s@strut}[true]{\eql@decide@if{#3}{#2}{#1}%
3296   {\let\eql@strut@field\eql@strut}{\let\eql@strut@field\relax}}
3297 \eql@define@key\eql@keycat{s@struttag}[true]{\eql@decide@if{#3}{#2}{#1}%
3298   {\let\eql@strut@tag\eql@strut}{\let\eql@strut@tag\relax}}
```

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

```
3299 \eql@define@key{equations}{displaybreak}[4]{\eql@displaybreak@pre{#1}}
3300 \def\eql@keycat{equations,setup}
3301 \eql@define@key\eql@keycat{allowbreaks,allowdisplaybreaks}[4]{%
3302   \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax}
3303 \eql@define@key\eql@keycat{displayheight}[\ht\eql@strutbox@]{%
3304   \def\eql@display@height{#1}}
3305 \eql@define@key\eql@keycat{displaydepth}[\dp\eql@strutbox@]{%
3306   \def\eql@display@depth{#1}}
```

**TODO:** describe

```
3307 \eql@define@key{equations}{noskip}[]{%
3308   \eql@decide@abovebelow{#3}{#2}{#1}%
3309   {\eql@display@atend{%
3310     \abovedisplayskip\z@skip
```

```

3311 \abovedisplayskip\z@skip
3312 \eql@abovedisplaytagskip@{z@skip}}%
3313 {\eql@display@atend{%
3314 \belowdisplayskip\z@skip
3315 \belowdisplayskip\z@skip
3316 \eql@belowdisplaytagskip@{z@skip}}}}
3317 \eql@define@key{equations}{short}[]{%
3318 \eql@decide@abovebelow{#3}{#2}{#1}%
3319 {\eql@display@atend{%
3320 \abovedisplayskip\abovedisplayskip}}}%
3321 {\eql@display@atend{%
3322 \belowdisplayskip\belowdisplayskip}}}}
3323 \eql@define@key{equations}{long}[]{%
3324 \eql@decide@abovebelow{#3}{#2}{#1}%
3325 {\eql@display@atend{%
3326 \abovedisplayskip\abovedisplayskip}}}%
3327 {\eql@display@atend{%
3328 \belowdisplayskip\belowdisplayskip}}}}
3329 \eql@define@key{equations}{medskip}[]{%
3330 \eql@decide@abovebelow{#3}{#2}{#1}%
3331 {\eql@display@atend{%
3332 \abovedisplayskip\eql@abovedisplaymedskip@}}}%
3333 {\eql@display@atend{%
3334 \belowdisplayskip\eql@belowdisplaymedskip@}}}}

```

**TODO:** describe

```

3335 \eql@define@key{equations}{skip}{%
3336 \setlength\eql@abovedisplayskip@{#1}%
3337 \eql@belowdisplayskip@{eql@abovedisplayskip@%
3338 \eql@display@atend{%
3339 \abovedisplayskip\eql@abovedisplayskip@
3340 \abovedisplayshortskip\abovedisplayskip
3341 \eql@abovedisplaytagskip@{abovedisplayskip
3342 \belowdisplayskip\abovedisplayskip
3343 \belowdisplayshortskip\belowdisplayskip
3344 \eql@belowdisplaytagskip@{belowdisplayskip}}}}
3345 \eql@define@key{equations}{aboveskip}{%
3346 \setlength\eql@abovedisplayskip@{#1}%
3347 \eql@display@atend{%
3348 \abovedisplayskip\eql@abovedisplayskip@
3349 \abovedisplayshortskip\abovedisplayskip
3350 \eql@abovedisplaytagskip@{abovedisplayskip}}}}
3351 \eql@define@key{equations}{belowskip}{%
3352 \setlength\eql@belowdisplayskip@{#1}%
3353 \eql@display@atend{%
3354 \belowdisplayskip\eql@belowdisplayskip@
3355 \belowdisplayshortskip\belowdisplayskip%
3356 \eql@belowdisplaytagskip@{belowdisplayskip}}}}
3357 \eql@define@key{equations}{abovespace}{%
3358 \addtolength\eql@abovespace@{#1}}
3359 \eql@define@key{equations}{belowspace}{%
3360 \addtolength\eql@belowspace@{#1}}

```

**TODO:** describe

```

3361 \eql@define@key{intertext}{skip}{%
3362 \setlength\abovedisplayskip{#1}\belowdisplayskip\abovedisplayskip}
3363 \eql@define@key{intertext}{aboveskip}{%
3364 \setlength\belowdisplayskip{#1}}

```

```

3365 \eql@define@key{intertext}{belowskip}{%
3366   \setlength\abovedisplayskip{#1}}
3367 \eql@define@key{intertext}{noskip}[]{%
3368   \eql@decide@abovebelow{#3}{#2}{#1}%
3369   {\belowdisplayskip\z@skip\belowdisplayshortskip\z@skip}%
3370   {\abovedisplayskip\z@skip\abovedisplayshortskip\z@skip}}
3371 \eql@define@key{intertext}{short}[]{%
3372   \eql@decide@abovebelow{#3}{#2}{#1}%
3373   {\belowdisplayskip\belowdisplayshortskip}%
3374   {\abovedisplayskip\abovedisplayshortskip}}
3375 \eql@define@key{intertext}{long}[]{%
3376   \eql@decide@abovebelow{#3}{#2}{#1}%
3377   {\belowdisplayshortskip\belowdisplayskip}%
3378   {\abovedisplayshortskip\abovedisplayskip}}
3379 \eql@define@key{intertext}{medskip}[]{%
3380   \eql@decide@abovebelow{#3}{#2}{#1}%
3381   {\belowdisplayskip\eql@belowdisplaymedskip@}%
3382   {\abovedisplayskip\eql@abovedisplaymedskip@}}

```

**TODO:** describe

```

3383 \eql@define@key{setup}{skip}{%
3384   \setlength\abovedisplayskip{#1}\belowdisplayskip\abovedisplayskip}
3385 \eql@define@key{setup}{aboveskip}{%
3386   \setlength\abovedisplayskip{#1}}
3387 \eql@define@key{setup}{belowskip}{%
3388   \setlength\belowdisplayskip{#1}}
3389 \eql@define@key{setup}{aboveshortskip}{%
3390   \setlength\abovedisplayshortskip{#1}}
3391 \eql@define@key{setup}{belowshortskip}{%
3392   \setlength\belowdisplayshortskip{#1}}
3393 \eql@define@key{setup}{tagskip}{%
3394   \setlength\eql@abovedisplaytagskip@{#1}%
3395   \eql@belowdisplaytagskip@\eql@abovedisplaytagskip@}
3396 \eql@define@key{setup}{abovetagskip}{%
3397   \setlength\eql@abovedisplaytagskip@{#1}}
3398 \eql@define@key{setup}{belowtagskip}{%
3399   \setlength\eql@belowdisplaytagskip@{#1}}
3400 \eql@define@key{setup}{medskip}{%
3401   \setlength\eql@abovedisplaymedskip@{#1}%
3402   \eql@belowdisplaymedskip@\eql@abovedisplaymedskip@}
3403 \eql@define@key{setup}{abovemedskip}{%
3404   \setlength\eql@abovedisplaymedskip@{#1}}
3405 \eql@define@key{setup}{abovetopskip}{%
3406   \setlength\eql@abovedisplaytopskip@{#1}}
3407 \eql@define@key{setup}{aboveparskip}{%
3408   \setlength\eql@abovedisplayparskip@{#1}}
3409 \eql@define@key{setup}{abovecontskip}{%
3410   \setlength\eql@abovedisplaycontskip@{#1}}
3411 \eql@define@key{setup}{belowmedskip}{%
3412   \setlength\eql@belowdisplaymedskip@{#1}}
3413 \eql@define@key{setup}{shortmode}{%
3414   \eql@decide@select{#3}{#2}{#1}{%
3415     {\off,never,no}{\eql@displayshortmode@z@},%
3416     {\above,neverbelow,notbelow,belowoff}{\eql@displayshortmode@ne}},%
3417     {\belowone,belowsingle}{\eql@displayshortmode@tw@},%
3418     {\belowall,always,on}{\eql@displayshortmode@thr@}}}}

```

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

```
3419 \def\eq@keycat{equations,subequations}
3420 \eq@define@key\eq@keycat{label}{\eq@blocklabel@set{#1}}
3421 \eq@define@key\eq@keycat{tag}{\eq@blocktag@set{#1}}
3422 \eq@define@key\eq@keycat{tag*}{\eq@blocktag@setstar{#1}}
```

**Tag Spacing.** **TODO:** describe

```
3423 \def\eq@keycat{equations,setup}
3424 \eq@define@key\eq@keycat{tagmargin}{\setlength\eq@tagmargin@{#1}}
3425 \eq@define@key\eq@keycat{tagmargin*}{\settowidth\eq@tagmargin@{#1}}
3426 \eq@define@key\eq@keycat{mintagsep}{\setlength\eq@tagsepmin@{#1}}
3427 \eq@define@key\eq@keycat{mintagwidth}{\setlength\eq@tagwidthmin@{#1}}
3428 \eq@define@key\eq@keycat{mintagwidth*}{\settowidth\eq@tagwidthmin@{#1}}
```

**Tag Layout.** **TODO:** describe

```
3429 \eq@define@key{setup}{taglayout}{\eq@tag@setmake{#1}}
3430 \eq@define@key{setup}{taglayout*}{\eq@tag@setmake@{#1}}
3431 \eq@define@key{setup}{tagform}{\eq@tag@setform#1}
3432 \eq@define@key{setup}{tagform*}{\eq@tag@setform@{#1}}
3433 \eq@define@key{setup}{subeqtemplate}{\def\eq@subequations@template{#1}}
3434 \eq@define@key{setup}{autolabel}[true]{%
3435   \eq@decide@bool{#3}{#2}{#1}\eq@autolabel}
3436 \eq@define@key{setup}{autotag}[true]{%
3437   \eq@decide@bool{#3}{#2}{#1}\eq@autotag}
```

**Equation Numbering.** **TODO:** describe

```
3438 \def\eq@keycat{equations,setup}
3439 \eq@define@key\eq@keycat{numberline,numline,n}[all]{%
3440   \eq@numbering@set{#1}}
3441 \eq@define@key\eq@keycat{nonumber,nn,*}[]{\let\eq@numbering@active\eq@false}
3442 \eq@define@key\eq@keycat{donumber,dn,!}[]{\let\eq@numbering@active\eq@true}
3443 \eq@define@key\eq@keycat{number,num}[true]{%
3444   \eq@decide@bool{#3}{#2}{#1}\eq@numbering@active}
3445 \eq@define@key\eq@keycat{tagsleft,leqno}[]{\let\eq@tagsleft\eq@true}
3446 \eq@define@key\eq@keycat{tagsright,reqno}[]{\let\eq@tagsleft\eq@false}
3447 \eq@define@key\eq@keycat{tags,eqno}{%
3448   \eq@decide@select{#3}{#2}{#1}{%
3449     {right,r}{\let\eq@tagsleft\eq@false}},%
3450     {left,l}{\let\eq@tagsleft\eq@true}}}
```

**Horizontal Layout.** **TODO:** describe

```
3451 \def\eq@keycat{equations,setup}
3452 \eq@define@key\eq@keycat{layout}{\eq@decide@select{#3}{#2}{#1}{%
3453   {{center,c}{\let\eq@flushleft\eq@false}},%
3454   {{left,l}{\let\eq@flushleft\eq@true}}}}
3455 \eq@define@key\eq@keycat{center}[]{\let\eq@flushleft\eq@false}
3456 \eq@define@key\eq@keycat{flushleft,left}[]{\let\eq@flushleft\eq@true}
3457 \eq@define@key\eq@keycat{leftmargin}{%
3458   \setlength\eq@flushleftmargin@{#1}}
3459 \eq@define@key\eq@keycat{minleftmargin}{%
3460   \setlength\eq@flushleftmarginmin@{#1}}
3461 \eq@define@key\eq@keycat{maxleftmargin}{%
```

```

3462 \setlength\eq\flushleftmarginmax@{#1}}
3463 \eq\define@key\eq\keycat{maxleftmargin*}[]{%
3464 \setlength\eq\flushleftmarginmax@{.5\maxdimen}}

```

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

```

3465 \def\eq\keycat{equations,setup}
3466 \eq\define@key\eq\keycat{marginbadness}{\eq\marginbadness@#1\relax}
3467 \eq\define@key\eq\keycat{maxbadness}{\eq\maxbadness@#1\relax}
3468 \eq\define@key\eq\keycat{mincolsep}{\def\eq\colsepmin@val{#1}}
3469 \eq\define@key\eq\keycat{maxcolsep}{\def\eq\colsepmax@val{#1}}
3470 \eq\define@key\eq\keycat{maxcolsep*}[]{\def\eq\colsepmax@val{.5\maxdimen}}
3471 \eq\define@key\eq\keycat{margins}[true]{%
3472 \eq\decide@bool{#3}{#2}{#1}\eq\align@margins}
3473 \def\eq\keycat{equationsbox,setup}
3474 \eq\define@key\eq\keycat{margin}{%
3475 \def\eq\box@marginleft{#1}\def\eq\box@marginright{#1}}
3476 \eq\define@key\eq\keycat{marginleft}{\def\eq\box@marginleft{#1}}
3477 \eq\define@key\eq\keycat{marginright}{\def\eq\box@marginright{#1}}

```

**Horizontal Shape.** **TODO:** describe

```

3478 \def\eq\keycat{equations,equationsbox,setup}
3479 \eq\define@key\eq\keycat{shape}[default]{\eq\shape@set{#1}}
3480 \eq\define@key\eq\keycat{padding,pad}[\eq\indent@val]{%
3481 \let\eq\paddingmax\eq>false}
3482 \def\eq\paddingleft@val{#1}\def\eq\paddingright@val{#1}}
3483 \eq\define@key\eq\keycat{padleft}[\eq\indent@val]{%
3484 \let\eq\paddingmax\eq>false\def\eq\paddingleft@val{#1}}
3485 \eq\define@key\eq\keycat{padright}[\eq\indent@val]{%
3486 \let\eq\paddingmax\eq>false\def\eq\paddingright@val{#1}}
3487 \eq\define@key\eq\keycat{padmax}[true]{%
3488 \eq\decide@bool{#3}{#2}{#1}\eq\paddingmax}
3489 \eq\define@key\eq\keycat{indent}[2em]{%
3490 \def\eq\indent@val{#1}}
3491 \eq\define@key\eq\keycat{indent*}[2em]{%
3492 \def\eq\indent@val{#1}\def\eq\paddingleft@val{#1}}

```

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

```

3493 \def\eq\keycat{equations,equationsbox,setup}
3494 \eq\define@key\eq\keycat{classout}{\eq\class@innerleft@set{#1}}
3495 \eq\define@key\eq\keycat{classin}{\eq\class@innerright@set{#1}}
3496 \eq\define@key\eq\keycat{classin*}{\eq\class@innerlead@set{#1}}
3497 \eq\define@key\eq\keycat{ampeq}[]{\eq\class@ampeq}
3498 \eq\define@key\eq\keycat{eqamp}[]{\eq\class@eqamp}
3499 \eq\define@key\eq\keycat{class}{\eq\decide@select{#3}{#2}{#1}{%
3500 {ampeq,amprel,eqafter,beforerel}\eq\class@ampeq},%
3501 {eqamp,relamp,eqbefore,afterrel}\eq\class@eqamp}}

```

**Punctuation.** **TODO:** describe

```

3502 \let\eq\punct@main\relax
3503 \def\eq\keycat{equations,equationsbox,setup}
3504 \eq\define@key\eq\keycat{punctsep}[\,]{\def\eq\punct@sep{#1}}
3505 \eq\define@key\eq\keycat{punct}[.]{\def\eq\punct@main{#1}}
3506 \eq\define@key\eq\keycat{punctline}[,]{\def\eq\punct@line{#1}}

```



```

3507 \eql@define@key\eql@keycat{punctcol}[,]{\def\eql@punct@col{#1}}
3508 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
3509 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
3510 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}

```

**Global Switches.** **TODO:** describe

```

3511 \let\eql@multi@linesfallback\eql@true
3512 \let\eql@single@crerror\eql@true
3513 \let\eql@ampproof@active\eql@true
3514 \eql@define@key{setup}{linesfallback}[true]{%
3515   \eql@decide@bool{#3}{#2}{#1}\eql@multi@linesfallback}
3516 \eql@define@key{setup}{ampproof}[true]{%
3517   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
3518 \eql@define@key{setup}{crerror}[true]{%
3519   \eql@decide@bool{#3}{#2}{#1}\eql@single@crerror}
3520 \eql@define@key{equations,setup}{rescan}[true]{%
3521   \eql@decide@if{#3}{#2}{#1}%
3522   {\let\eql@scan@body\eql@scan@body@rescan}%
3523   {\let\eql@scan@body\eql@scan@body@dump}}

```

**Package Options.** **TODO:** describe

```

3524 \let\eql@provide@amsmath\eql@true
3525 \let\eql@provide@equation\eql@true
3526 \let\eql@provide@ang\eql@true
3527 \let\eql@amsmath@dofixends\eql@true
3528 \eql@define@key{setup}{equation}[true]{%
3529   \eql@error@packageoption{#2}%
3530   \eql@decide@bool{#3}{#2}{#1}\eql@provide@equation}
3531 \eql@define@key{setup}{amsmath}[true]{%
3532   \eql@error@packageoption{#2}%
3533   \eql@decide@bool{#3}{#2}{#1}\eql@provide@amsmath}
3534 \eql@define@key{setup}{amsmathends}[true]{%
3535   \eql@error@packageoption{#2}%
3536   \eql@decide@bool{#3}{#2}{#1}\eql@amsmath@dofixends}
3537 \eql@define@key{setup}{ang}[true]{%
3538   \eql@error@packageoption{#2}%
3539   \eql@decide@bool{#3}{#2}{#1}\eql@provide@ang}

```

## P.4 Component Selection

**TODO:** describe

```

3540 \newenvironment{eql@gathered}
3541   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
3542 \newenvironment{eql@multlined}
3543   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
3544 \newenvironment{eql@aligned}
3545   {\eqnaddopt{align}\equationsbox}{\endequationsbox}

```

**TODO:** describe

```

3546 \newenvironment{eql@equation}
3547   {\eqnaddopt{equation}\equations}{\endequations}
3548 \newenvironment{eql@gather}
3549   {\eqnaddopt{lines}\equations}{\endequations}
3550 \newenvironment{eql@multline}

```

```

3551 {\eqnaddopt{lines,padmax,shape=steps,numberline=out}\equations}
3552 {\endequations}
3553 \newenvironment{eql@align}
3554 {\eqnaddopt{align}\equations}{\endequations}
3555 \newenvironment{eql@flalign}
3556 {\eqnaddopt{align,margins=false}\equations}{\endequations}
3557 \newenvironment{eql@equation*}
3558 {\eqnaddopt{nonumber}\eql@equation}{\endequations}
3559 \newenvironment{eql@gather*}
3560 {\eqnaddopt{nonumber}\eql@gather}{\endequations}
3561 \newenvironment{eql@multline*}
3562 {\eqnaddopt{nonumber}\eql@multline}{\endequations}
3563 \newenvironment{eql@align*}
3564 {\eqnaddopt{nonumber}\eql@align}{\endequations}
3565 \newenvironment{eql@flalign*}
3566 {\eqnaddopt{nonumber}\eql@flalign}{\endequations}

```

**TODO:** describe

```

3567 \def\eql@provide@movecmd#1#2{%
3568 \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
3569 }
3570 \def\eql@provide@undefinecmd#1{%
3571 \expandafter\let\csname #1\endcsname\@undefined
3572 }
3573 \def\eql@provide@moveenv#1#2{%
3574 \expandafter\let\csname #1\expandafter\endcsname\csname #2\endcsname
3575 \expandafter\let\csname end#1\expandafter\endcsname\csname end#2\endcsname
3576 }
3577 \def\eql@provide@undefineenv#1{%
3578 \expandafter\let\csname #1\endcsname\@undefined
3579 \expandafter\let\csname end#1\endcsname\@undefined
3580 }

```

**TODO:** describe

```

3581 \def\eql@provide@onlyonce#1#2{%
3582 \def\eql@tmp{#2}%
3583 \def\@tempa{#1}%
3584 \ifx\eql@tmp\@tempa
3585 \let\eql@tmp\@undefined
3586 \fi
3587 \ifx\eql@tmp\@empty
3588 \let\eql@tmp\@undefined
3589 \fi
3590 \def\@tempa{*}%
3591 \ifx\eql@tmp\@tempa
3592 \def\eql@tmp{#1}%
3593 \fi
3594 \ifdefined\eql@tmp\else
3595 \ifcsname eql@provided@#1\endcsname
3596 \def\eql@tmp{#1}%
3597 \else
3598 \expandafter\let\csname eql@provided@#1\endcsname\eql@true
3599 \fi
3600 \fi
3601 }

```

**TODO:** describe

```

3602 \def\eql@provide@cmdonlyonce#1#2{%

```

```

3603 \def\eql@tmp{#2}%
3604 \edef\@tempb{\expandafter\noexpand\csname#1\endcsname}%
3605 \ifx\eql@tmp\@tempb
3606   \let\eql@tmp\@undefined
3607 \fi
3608 \ifx\eql@tmp\@empty
3609   \let\eql@tmp\@undefined
3610 \fi
3611 \def\@tempa{*}%
3612 \ifx\eql@tmp\@tempa
3613   \let\eql@tmp\@tempb
3614 \fi
3615 \ifdefined\eql@tmp
3616   \edef\eql@tmp{\expandafter\expandafter\expandafter\@gobble
3617     \expandafter\string\eql@tmp}%
3618 \else
3619   \ifcsname eql@provided@#1\endcsname
3620     \let\eql@tmp\@tempb
3621   \else
3622     \expandafter\let\csname eql@provided@#1\endcsname\eql@true
3623   \fi
3624 \fi
3625 }

```

**TODO:** describe

```

3626 \def\eql@provide@cmd#1#2{%
3627   \eql@provide@cmdonlyonce{#1}{#2}%
3628   \ifdefined\eql@tmp
3629     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@#1}%
3630   \else
3631     \eql@amsmath@after{%
3632       \eql@provide@movecmd{ams#1}{#1}%
3633       \eql@provide@movecmd{#1}{eql@#1}%
3634     }%
3635     \AddToHook{package/mathtools/after}{%
3636       \eql@provide@movecmd{#1}{eql@#1}%
3637     }%
3638     \eql@provide@movecmd{#1}{eql@#1}%
3639     \eql@amsmath@futurebefore{\eql@provide@undefinecmd{#1}}%
3640   \fi
3641 }

```

**TODO:** describe

```

3642 \def\eql@amsmath@endfix#1#2{%
3643   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
3644   \long\edef\@tempb{\expandafter\noexpand\csname eql@amsmath@end#2\endcsname}%
3645   \expandafter\ifx\csname end#1\endcsname\@tempa
3646     \expandafter\let\csname end#1\endcsname\@tempb
3647   \fi
3648 }

```

**TODO:** describe

```

3649 \def\eql@amsmath@fixends{%
3650   \eql@amsmath@after{%
3651     \let\eql@amsmath@endmultiline\endmultiline
3652     \eql@amsmath@endfix{multiline*}{multiline}%
3653     \let\eql@amsmath@endgather\endgather
3654     \eql@amsmath@endfix{gather*}{gather}%

```

```

3655 \let\eql@amsmath@endalign\endalign
3656 \eql@amsmath@endfix{align*}{align}%
3657 \eql@amsmath@endfix{flalign}{align}%
3658 \eql@amsmath@endfix{flalign*}{align}%
3659 \eql@amsmath@endfix{alignat}{align}%
3660 \eql@amsmath@endfix{alignat*}{align}%
3661 \eql@amsmath@endfix{xalignat}{align}%
3662 \eql@amsmath@endfix{xalignat*}{align}%
3663 \eql@amsmath@endfix{xxalignat}{align}%
3664 \let\eql@amsmath@endaligned\endaligned
3665 \eql@amsmath@endfix{gathered}{aligned}%
3666 \eql@amsmath@endfix{alignedat}{aligned}%
3667 }
3668 }

```

**TODO:** describe

```

3669 \def\eql@provide@env#1#2{%
3670 \eql@provide@onlyonce{#1}{#2}%
3671 \ifdefined\eql@tmp
3672 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@#1}%
3673 \else
3674 \eql@amsmath@after{%
3675 \eql@provide@moveenv{ams#1}{#1}%
3676 \eql@provide@moveenv{ams#1*}{#1*}%
3677 \eql@provide@moveenv{#1}{\eql@#1}%
3678 \eql@provide@moveenv{#1*}{\eql@#1*}%
3679 }%
3680 \AddToHook{package/mathtools/after}{%
3681 \eql@provide@moveenv{#1}{\eql@#1}%
3682 \eql@provide@moveenv{#1*}{\eql@#1*}%
3683 }%
3684 \eql@provide@moveenv{#1}{\eql@#1}%
3685 \eql@provide@moveenv{#1*}{\eql@#1*}%
3686 \eql@amsmath@futurebefore{\eql@provide@undefineenv{#1}}%
3687 \eql@amsmath@futurebefore{\eql@provide@undefineenv{#1*}}%
3688 \fi
3689 }

```

**TODO:** describe

```

3690 \def\eql@provide@env@equation#1{%
3691 \eql@provide@onlyonce{equation}{#1}%
3692 \ifdefined\eql@tmp
3693 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{\eql@equation}%
3694 \else
3695 \eql@amsmath@after{%
3696 \eql@provide@moveenv{amsequation}{equation}%
3697 \eql@provide@moveenv{amsequation*}{equation*}%
3698 \eql@provide@moveenv{equation}{\eql@equation}%
3699 \eql@provide@moveenv{equation*}{\eql@equation*}%
3700 }%
3701 \AddToHook{package/hyperref/after}{%
3702 \@ifpackageloaded{amsmath}{}%
3703 \let\latexequation\H@equation
3704 \let\endlatexequation\H@endequation
3705 \eql@provide@moveenv{hyperrefequation}{equation}%
3706 \eql@provide@moveenv{equation}{\eql@equation}%
3707 }%
3708 }%

```

```

3709 \ifpackage{amsmath}{\ifpackage{hyperref}}{%
3710 \eql@provide@moveenv{latexequation}{equation}%
3711 }}%
3712 \eql@provide@moveenv{equation}{eql@equation}%
3713 \eql@provide@moveenv{equation*}{eql@equation*}%
3714 \eql@amsmath@futurebefore{\eql@provide@undefineenv{equation*}}%
3715 \fi
3716 }

```

**TODO:** describe

```

3717 \def\eql@provide@env@multlined#1{%
3718 \eql@provide@onlyonce{multlined}{#1}%
3719 \ifdefined\eql@tmp
3720 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
3721 \else
3722 \AddToHook{package/mathtools/after}{%
3723 \eql@provide@moveenv{amsmultlined}{multlined}%
3724 \eql@provide@moveenv{multlined}{eql@multlined}%
3725 }%
3726 \eql@provide@moveenv{multlined}{eql@multlined}%
3727 \ifpackage{mathtools}{\AddToHook{package/mathtools/before}{%
3728 \eql@provide@undefineenv{multlined}}}%
3729 \fi
3730 }

```

**TODO:** describe

```

3731 \def\eql@provide@env@subequations@reinstall{%
3732 \global\let\subequations\eql@subequations
3733 \let\@tempa\UseHook
3734 \let\@tempb\UseHookWithArguments
3735 \def\UseHook{\let\UseHook\@tempa\let\UseHookWithArguments\@tempb%
3736 \expandafter\subequations@gobble}%
3737 \def\UseHookWithArguments{\expandafter\UseHook@gobble}%
3738 }

```

**TODO:** describe

```

3739 \def\eql@provide@env@subequations#1{%
3740 \eql@provide@onlyonce{subequations}{#1}%
3741 \ifdefined\eql@tmp
3742 \expandafter\eql@provide@moveenv
3743 \expandafter{\eql@tmp}{eql@subequations}%
3744 \else
3745 \eql@amsmath@after{%
3746 \eql@provide@moveenv{amssubequations}{subequations}%
3747 \eql@provide@moveenv{subequations}{eql@subequations}%
3748 \let\subequations\eql@provide@env@subequations@reinstall
3749 }%
3750 \AddToHook{package/hyperref/after}{%
3751 \AddToHook{cmd/subequations/before}[hyperref]}%
3752 \AddToHook{cmd/subequations/after}[hyperref]}%
3753 \RemoveFromHook{cmd/subequations/before}[hyperref]%
3754 \RemoveFromHook{cmd/subequations/after}[hyperref]%
3755 \AddToHook{cmd/amssubequations/before}%
3756 {%
3757 \stepcounter{equation}%
3758 \protected@edef\theHparentequation{\theHequation}%
3759 \addtocounter{equation}{-1}%
3760 }

```

```

3761     \AddToHook{cmd/amssubequations/after}%
3762     {%
3763         \def\theHequation{\theHparentequation\alph{equation}}%
3764         \ignorespaces
3765     }
3766 }
3767 \eql@provide@moveenv{subequations}{eql@subequations}%
3768 \AddToHook{cmd/subequations/after}{}%
3769 \let\subequations\eql@provide@env@subequations@reinstall
3770 \eql@amsmath@futurebefore{\eql@provide@undefineenv{subequations}}%
3771 \fi
3772 }

```

**TODO:** describe

```

3773 \def\eql@provide@sqr{%
3774     \let\[\eql@equations@sqr@open
3775     \let\]\eql@equations@sqr@close
3776     \eql@amsmath@after{%
3777         \let\[\eql@equations@sqr@open
3778         \let\]\eql@equations@sqr@close
3779     }%
3780 }

```

**TODO:** describe

```

3781 \def\eql@provide@ang{%
3782     \let\<\eql@equations@ang@open
3783     \let\>\eql@equations@ang@close
3784 }

```

**TODO:** describe

```

3785 \eql@define@key{provide}{equation}[]{\eql@provide@env@equation{#1}}
3786 \eql@define@key{provide}{gather}[]{\eql@provide@env{gather}{#1}}
3787 \eql@define@key{provide}{multline}[]{\eql@provide@env{multline}{#1}}
3788 \eql@define@key{provide}{align}[]{\eql@provide@env{align}{#1}}
3789 \eql@define@key{provide}{flalign}[]{\eql@provide@env{flalign}{#1}}
3790 \eql@define@key{provide}{aligned}[]{\eql@provide@env{aligned}{#1}}
3791 \eql@define@key{provide}{gathered}[]{\eql@provide@env{gathered}{#1}}
3792 \eql@define@key{provide}{multlined}[]{\eql@provide@env@multlined{#1}}
3793 \eql@define@key{provide}{subequations}[]{\eql@provide@env@subequations{#1}}
3794 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
3795 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
3796 \eql@define@key{provide}{eqref}[]{\eql@provide@cmd{eqref}{#1}}
3797 \eql@define@key{provide}{notag}[]{\eql@provide@cmd{notag}{#1}}
3798 \eql@define@key{provide}{thetag}[]{\eql@provide@cmd{thetag}{#1}}
3799 \eql@define@key{provide}{allowdisplaybreaks}[]{%
3800     \eql@provide@cmd{allowdisplaybreaks}{#1}}
3801 \eql@define@key{provide}{numberwithin}[]{\eql@provide@cmd{numberwithin}{#1}}

```

**TODO:** describe

```

3802 \newcommand{\eqnlinesprovide}[1]{%
3803 (dev)\eql@dev@start\eqnlinesprovide
3804     \eql@setkeys{provide}{#1}}

```

## P.5 Global and Package Options

Handle global and package options:

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

```
3805 \newcommand{\eqnlineset}[1]{%
3806 (dev)\eql@dev@start\eqnlineset
3807 \eql@setkeys{setup}{#1}}
```

Disable error message for exclusive package options:

```
3808 \let\eql@error@packageoption\@gobble
```

Pass undeclared options on to keyval processing:

```
3809 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Process package options:

```
3810 \ProcessOptions
```

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

```
3811 \def\eql@error@packageoption#1{%
3812 \eql@error{may only use ‘#1’ as a package option}%
3813 }
```

Provide classes of interfaces:

```
3814 \ifcsname tagsleft@true\endcsname\else
3815 \expandafter\newif\csname iftagsleft@\endcsname
3816 \fi
3817 \ifcsname @fleqntrue\endcsname\else
3818 \expandafter\newif\csname if@fleqn\endcsname
3819 \fi
3820 \ifdefined\eql@provide@amsmath
3821 \let\eql@provide@equation\eql@true
3822 \eql@amsmath@after{%
3823 \iftagsleft@
3824 \eqnlineset{tagsleft}
3825 \else
3826 \eqnlineset{tagsright}
3827 \fi
3828 \if@fleqn
3829 \eqnlineset{left}
3830 \else
3831 \eqnlineset{center}
3832 \fi
3833 }
3834 \fi
```

**TODO:** describe

```
3835 \ifdefined\eql@provide@equation
3836 \eqnlinesprovide{equation,sqr}
3837 \fi
```

**TODO:** describe

```
3838 \ifdefined\eql@amsmath@dofixends
3839 \eql@amsmath@fixends
3840 \fi
```

**TODO:** describe

```
3841 \ifdefined\eqL@provide@amsmath
3842   \eqnlinesprovide{%
3843     multiline,gather,align,flalign,%
3844     multlined,gathered,aligned,%
3845     subequations,eqref}
3846 \fi
```

**TODO:** describe

```
3847 \ifdefined\eqL@provide@ang
3848   \eqnlinesprovide{ang}
3849 \fi
```