

# The eqnlines Package

Niklas Beisert

Institut für Theoretische Physik  
Eidgenössische Technische Hochschule Zürich  
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland

`nbeisert@itp.phys.ethz.ch`

2025/05/18, v0.9

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

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

## Abstract

`eqnlines` is a  $\text{\LaTeX} 2_{\epsilon}$  package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of  $\text{\LaTeX}$  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 . . . . .                             | 8         |
| 2.3      | Horizontal Placement . . . . .                  | 11        |
| 2.4      | Punctuation . . . . .                           | 15        |
| 2.5      | Math Classes at Alignment . . . . .             | 16        |
| 2.6      | Vertical Spacing . . . . .                      | 17        |
| 2.7      | Further Environments and Features . . . . .     | 20        |
| 2.8      | General Options . . . . .                       | 22        |
| 2.9      | Feature Selection and Package Options . . . . . | 22        |
| <b>3</b> | <b>Information</b>                              | <b>23</b> |
| 3.1      | Copyright . . . . .                             | 23        |
| 3.2      | Credits . . . . .                               | 23        |
| 3.3      | Files and Installation . . . . .                | 24        |
| 3.4      | Related CTAN Packages . . . . .                 | 24        |
| 3.5      | Feature Suggestions . . . . .                   | 25        |
| 3.6      | Revision History . . . . .                      | 25        |
| <b>A</b> | <b>Implementation</b>                           | <b>27</b> |

|          |   |           |
|----------|---|-----------|
| <b>B</b> | <b>General Support</b>                    | <b>27</b> |
| B.1      | Development Messages . . . . .            | 27        |
| B.2      | Supporting Definitions . . . . .          | 27        |
| B.3      | Dollardollar Abstraction . . . . .        | 28        |
| B.4      | Look-Ahead in Alignment . . . . .         | 28        |
| B.5      | Error Messages . . . . .                  | 31        |
| B.6      | amsmath Integration . . . . .             | 31        |
| B.7      | PDF Tagging Support . . . . .             | 32        |
| B.8      | Key-Value Processing . . . . .            | 33        |
| <b>C</b> | <b>Parameters and Registers</b>           | <b>35</b> |
| C.1      | Parameters . . . . .                      | 36        |
| C.2      | Registers . . . . .                       | 38        |
| C.3      | Hooks . . . . .                           | 40        |
| <b>D</b> | <b>Features</b>                           | <b>41</b> |
| D.1      | Punctuation . . . . .                     | 41        |
| D.2      | Math Classes at Alignment . . . . .       | 42        |
| D.3      | Framed Cells . . . . .                    | 43        |
| D.4      | Alternative Content Description . . . . . | 44        |
| <b>E</b> | <b>Equation Numbering</b>                 | <b>44</b> |
| E.1      | Supporting Definitions . . . . .          | 44        |
| E.2      | Schemes . . . . .                         | 45        |
| E.3      | Interface . . . . .                       | 48        |
| E.4      | Integration . . . . .                     | 52        |
| E.5      | Component Display . . . . .               | 55        |
| E.6      | Tag Composition . . . . .                 | 58        |
| <b>F</b> | <b>Subequation Numbering</b>              | <b>60</b> |
| F.1      | Definitions . . . . .                     | 60        |
| F.2      | Environment . . . . .                     | 61        |
| F.3      | Subequation Scheme . . . . .              | 63        |
| <b>G</b> | <b>Display Equations Support</b>          | <b>63</b> |
| G.1      | Display Breaks . . . . .                  | 64        |
| G.2      | Explicit Vertical Space . . . . .         | 65        |
| G.3      | Default Vertical Spacing . . . . .        | 66        |
| G.4      | Entry and Exit . . . . .                  | 66        |
| G.5      | Stack . . . . .                           | 72        |
| <b>H</b> | <b>Multi-Line Support</b>                 | <b>73</b> |
| H.1      | Measure Support . . . . .                 | 73        |
| H.2      | Line Breaks . . . . .                     | 74        |
| H.3      | Intertext . . . . .                       | 75        |
| <b>I</b> | <b>Column Placement</b>                   | <b>77</b> |
| I.1      | Supporting Definitions . . . . .          | 77        |
| I.2      | Shape Schemes . . . . .                   | 78        |
| I.3      | Width Data . . . . .                      | 81        |
| I.4      | Best Line Selection . . . . .             | 84        |
| I.5      | Tag Margin . . . . .                      | 85        |
| I.6      | Single Column . . . . .                   | 86        |
| I.7      | Multiple Columns . . . . .                | 87        |

|          |   |            |
|----------|---|------------|
| <b>J</b> | <b>Single Column Arrangement</b>        | <b>93</b>  |
| J.1      | Supporting Definitions . . . . .        | 93         |
| J.2      | Arrangement Methods . . . . .           | 93         |
| J.3      | Central Alignment . . . . .             | 95         |
| J.4      | Left Alignment . . . . .                | 96         |
| J.5      | Right Alignment . . . . .               | 97         |
| <b>K</b> | <b>Equations Box Environment</b>        | <b>98</b>  |
| K.1      | Line Breaks . . . . .                   | 99         |
| K.2      | Stacked Mode . . . . .                  | 99         |
| K.3      | Aligned Mode . . . . .                  | 100        |
| K.4      | Main . . . . .                          | 102        |
| K.5      | Environment . . . . .                   | 103        |
| <b>L</b> | <b>Single-Line Equation</b>             | <b>104</b> |
| L.1      | Native Mode . . . . .                   | 104        |
| L.2      | Print . . . . .                         | 105        |
| <b>M</b> | <b>Multi-Line with Single Column</b>    | <b>106</b> |
| M.1      | Measure . . . . .                       | 106        |
| M.2      | Column Placement . . . . .              | 108        |
| M.3      | Print . . . . .                         | 108        |
| <b>N</b> | <b>Multi-Line with Multiple Columns</b> | <b>109</b> |
| N.1      | Support . . . . .                       | 109        |
| N.2      | Transpose . . . . .                     | 110        |
| N.3      | Measure . . . . .                       | 112        |
| N.4      | Columns Placement . . . . .             | 114        |
| N.5      | Print . . . . .                         | 115        |
| <b>O</b> | <b>Interface</b>                        | <b>118</b> |
| O.1      | Scanning the Equation Body . . . . .    | 118        |
| O.2      | Options Processing . . . . .            | 121        |
| O.3      | Single-Line Main . . . . .              | 123        |
| O.4      | Multi-Line Main . . . . .               | 124        |
| O.5      | Equations Environment . . . . .         | 126        |
| <b>P</b> | <b>Options</b>                          | <b>128</b> |
| P.1      | Selection Tools . . . . .               | 128        |
| P.2      | Options Declarations . . . . .          | 129        |
| P.3      | Parameter Presets . . . . .             | 139        |
| P.4      | Component Selection . . . . .           | 140        |
| P.5      | Global and Package Options . . . . .    | 149        |

## 1 Introduction

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

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

layout where needed. The main additions are as follows:

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

The package represents a stand-alone implementation of an equations environment which is largely compatible with the established  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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 v0.9:** Please note that this package is still in a development and testing stage in the present version. This mainly applies to the documentation

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

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

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

To use the `eqnlines` package add the command

$$\backslash usepackage{eqnlines}$$

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

## 2.1 Equations Environment

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

$$\begin{array}{l} \backslash begin{equations} mod [opts] mod\_ \\ \dots \\ \backslash end{equations} \end{array}$$

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

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

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

$$\backslash eqnlineset{opts}$$

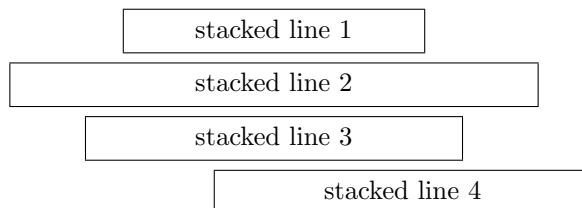
`\eqncontrol` Several options or features can be controlled for individual lines or cells by the macro:

$$\backslash eqncontrol{opts}$$

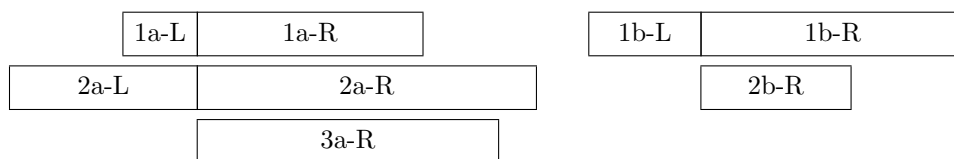
**Modes of Operation.** The package supplies a main maths environment called `equations` which has three principal modes of operation. It can display a single-line equation just as the  $\text{\LaTeX}$  environment `equation` or the symbolic shortcut `\[...\]`:

|             |
|-------------|
| single line |
|-------------|

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



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



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

`lines` (*key*)

`columns` (*key*)

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

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

---

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

---

<sup>1</sup>Arguably, a single-line equation is just a stack of equations of height 1. Nevertheless, there is a single-line mode which prohibits line breaks and which works slightly more efficiently: For example, the multi-line modes will process the input twice which is not needed for the single-line mode. Apart from that, the package takes care that the layout and spacing of single-line equations and multi-line equations consisting of a single line is the same.

`\[...]` **Alternative Forms.** The package offers several alternative names for the same mode as `\<...>` well as a symbolic short form `\<...>` extending the L<sup>A</sup>T<sub>E</sub>X display equation form `\[...]` to multi-line equations. An additional equal sign ‘=’ in `\<=...>` serves as a modifier character which acts as a short form for the optional argument `lines` selecting the lines mode. Similarly, the modifiers minus ‘-’ and bar ‘|’ select single-line and columns mode, `sqropt` (key) respectively. Both short forms can be customised by setting default arguments via the global `angopt` (key) options `sqropt={opts}` and `angopt={opts}`. Both default arguments are preset to `nonumber` which disables equation numbering, see section 2.2.

---

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

---

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

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

---

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

---

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

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

---

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

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

---

## 2.2 Numbering

`numberline` (*key*) **Numbering Schemes.** The package extends the established interface of L<sup>A</sup>T<sub>E</sub>X and the `amsmath` package for labelling equations with numbers or with manually assigned tags. For multi-line equations, there are two distinct modes of operations: individual labelling of the equation lines or one overall number/tag for the whole block of equations. The modes are selected by an optional argument `numberline=mode` (alternatively `nline` or just `n`) as follows:

| name                | alt.           | description      | preset                                     |
|---------------------|----------------|------------------|--|
| <code>all</code>    | <code>a</code> | individual lines | all lines                                  |
| <code>sub</code>    | <code>s</code> |                  | subequations (a, b, c, ...)                |
| <code>multi</code>  | <code>@</code> |                  | individual lines, numbering on             |
| <code>none</code>   | <code>-</code> |                  | individual lines, numbering off            |
| <code>first</code>  | <code>f</code> | single number    | first line                                 |
| <code>last</code>   | <code>l</code> |                  | last line                                  |
| <code>middle</code> | <code>m</code> |                  | 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>center</code> | <code>c</code> |                  | vertically centred (not yet implemented)   |
| <code>here</code>   | <code>h</code> |                  | line indicated by <code>\numberhere</code> |
| <code>best</code>   | <code>+</code> |                  | line with most available space             |
| <code>single</code> | <code>1</code> |                  | previous single-line mode, numbering on    |
| <code>on</code>     | <code>!</code> | activation       | turn numbering on                          |
| <code>off</code>    | <code>*</code> |                  | turn numbering off                         |

---

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



|                      |                         |                          |
|----------------------|-------------------------|--------------------------|
| <b>none:</b>         | <b>all:</b>             | <b>sub:</b>              |
| $x = \cos \phi$      | $x = \cos \phi$ (12)    | $x = \cos \phi$ (16a)    |
| $= (z + z^{-1})/2$   | $= (z + z^{-1})/2$ (13) | $= (z + z^{-1})/2$ (16b) |
| $\phi = \arccos x$   | $\phi = \arccos x$ (14) | $\phi = \arccos x$ (16c) |
| $= -i \log z$        | $= -i \log z$ (15)      | $= -i \log z$ (16d)      |
| <b>first:</b>        | <b>middle:</b>          | <b>last:</b>             |
| $x = \cos \phi$ (17) | $x = \cos \phi$         | $x = \cos \phi$ (20)     |
| $= (z + z^{-1})/2$   | $= (z + z^{-1})/2$      | $= (z + z^{-1})/2$       |
| $\phi = \arccos x$   | $\phi = \arccos x$ (18) | $\phi = \arccos x$       |
| $= -i \log z$        | $= -i \log z$           | $= -i \log z$ (19)       |

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

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

**\nonumber**      and      **\donumber**

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

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

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

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

**\numbernext**      **\numberhere**      and      **\numbernext**

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

---

```

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

```

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

(21)

```

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

```

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

(23)

```

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

```

$$\begin{aligned}
 x &= \cos \phi \\
 \phi &= \arccos x
 \end{aligned}$$

(25)

|   |  |
|---|--|
| <code>\eqnlineset{angopt=donumber}</code>                                 |  |
| <code>\&lt;* x &amp;= \cos\phi \\\phi &amp;= \arccos x \&gt;</code>       | $x = \cos \phi$<br>$\phi = \arccos x$      |
| <code>\begin{equations}</code>  |  |
| <code>  x &amp;= \cos\phi \numbernext \\\  &amp;= (z+z^{-1})/2 \\\</code> | $x = \cos \phi$<br>$= (z + z^{-1})/2$ (26) |
| <code>\phi &amp;= \arccos x \numbernext \\\  &amp;= -i\log z</code>       | $\phi = \arccos x$<br>$= -i \log z$ (27)   |
| <code>\end{equations}</code>  |  |
| <code>\eqnlineset{numberline=here}</code>                                 |  |
| <code>\&lt;!</code>   | $x = \cos \phi$                            |
| <code>  x &amp;= \cos\phi \\\  &amp;= (z+z^{-1})/2 \\\</code>             | $= (z + z^{-1})/2$                         |
| <code>\phi &amp;= \arccos x \numberhere \\\  &amp;= -i\log z</code>       | $\phi = \arccos x$ (28)                    |
| <code>\&gt;</code>  | $= -i \log z$                              |
| <code>\eqnlineset{numberline=first}</code>                                |  |
| <code>\&lt;!</code>   | $x = \cos \phi$                            |
| <code>  x &amp;= \cos\phi \numbernext \\\  &amp;= (z+z^{-1})/2 \\\</code> | $= (z + z^{-1})/2$ (29)                    |
| <code>\phi &amp;= \arccos x \numbernext \\\  &amp;= -i\log z</code>       | $\phi = \arccos x$                         |
| <code>\&gt;</code>  | $= -i \log z$                              |

---

`\label` **Labels and Tags.** Equation numbers can receive L<sup>A</sup>T<sub>E</sub>X labels as usual, and they can be `\tag` turned into manually assigned tags using the established macros:

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

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

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

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

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

`\[@{label}... \]`

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

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

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

`\eqref{label}`.

`\tagform` For custom typesetting, `\tagform` encloses a number/tag with decoration, `\tagbox` puts the decorated number in a box and `\tagboxed` combines the two.

`\tagboxed` The typesetting of equation numbers and tags passes through two macros, one which defines the layout and another one which adds a decoration by parentheses. These two methods can be adjusted via the options:

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

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

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

---

```

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

```

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

[30]

---

## 2.3 Horizontal Placement

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

`left (key)`

---

```

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

```

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

---

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

`tagtright (key)`



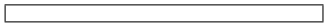

---

|  |                    |      |
|--|--------------------|------|
| <code>\&lt;[tags=right,!]</code>       |                    |      |
| <code>x &amp;= \cos\phi \\\</code>     | $x = \cos \phi$    | (31) |
| <code>&amp;= (z+z^{-1})/2 \\\</code>   | $= (z + z^{-1})/2$ | (32) |
| <code>\phi &amp;= \arccos x \\\</code> | $\phi = \arccos x$ | (33) |
| <code>&amp;= -i\log z</code>           | $= -i \log z$      | (34) |
| <code>\&gt;</code>                     |                    |      |
| <code>\&lt;[tags=left,!]</code>        |                    |      |
| <code>x &amp;= \cos\phi \\\</code>     | $x = \cos \phi$    | (35) |
| <code>&amp;= (z+z^{-1})/2 \\\</code>   | $= (z + z^{-1})/2$ | (36) |
| <code>\phi &amp;= \arccos x \\\</code> | $\phi = \arccos x$ | (37) |
| <code>&amp;= -i\log z</code>           | $= -i \log z$      | (38) |
| <code>\&gt;</code>                     |                    |      |

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

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

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

|  |   |      |
|--|---|------|
| <code>\eqnlineset{tagmarginthreshold=0.7}</code> |   |      |
| <code>\[! \framebox[4em]{} \]</code>             |  | (39) |
| <code>\[! \framebox[8em]{} \]</code>             |   | (40) |
| <code>\[! \framebox[12em]{} \]</code>            |   | (41) |
| <code>\[! \framebox[16em]{} \]</code>            |   | (42) |

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

---

```

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

```

---

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

---

```

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

```

---

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

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

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

---

|   |                      |                      |
|---|----------------------|----------------------|
| $\begin{aligned} &\backslash eqnlineset\{pad=2em\} \\ &\backslash <=[shape=...] \ x = \cos\phi \ \backslash \ x = (z+z^{-1})/2 \ \backslash \\ &\quad \backslash \phi = \arccos x \ \backslash \ \backslash \phi = -i\log z \ \backslash > \end{aligned}$ |                      |                      |
| <b>left:</b>  | <b>center:</b>       | <b>right:</b>        |
| $x = \cos \phi$   | $x = \cos \phi$      | $x = \cos \phi$      |
| $x = (z + z^{-1})/2$  | $x = (z + z^{-1})/2$ | $x = (z + z^{-1})/2$ |
| $\phi = \arccos x$  | $\phi = \arccos x$   | $\phi = \arccos x$   |
| $\phi = -i \log z$  | $\phi = -i \log z$   | $\phi = -i \log z$   |
| <b>first:</b>   | <b>hanging:</b>      | <b>steps:</b>        |
| $x = \cos \phi$   | $x = \cos \phi$      | $x = \cos \phi$      |
| $x = (z + z^{-1})/2$  | $x = (z + z^{-1})/2$ | $x = (z + z^{-1})/2$ |
| $\phi = \arccos x$  | $\phi = \arccos x$   | $\phi = \arccos x$   |
| $\phi = -i \log z$  | $\phi = -i \log z$   | $\phi = -i \log z$   |

---

**\shoveleft** The alignment preset can be adjusted for individual lines by the macros:  
**\shovecenter**  
**\shoveright**  $\backslash shoveleft | \backslash shovecenter | \backslash shoveright [*] | [dimen]$ ,

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 macros accept an optional argument **indent** (*key*) [*dimen*] specifying a variable amount of shift. They also accept the modifiers ‘\*’ or ‘!’ for indentation or hanging indentation by the standard indentation amount (**indent=2em**).  
**\shoveby** Furthermore, **\shoveby** [\*]{*dimen*} shifts the line by the additional amount *dimen* (the star variant shifts to an absolute position relative to the reference position).

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

---

|  |        |
|--|--------|
| $\backslash eqnlineset\{indent=2em,pad=5em\}$                                  |        |
| $\backslash <=$  |        |
| $\backslash shoveleft \quad \backslash framebox[5em]\{left\} \ \backslash$     | left   |
| $\backslash shoveleft* \quad \backslash framebox[5em]\{indent\} \ \backslash$  | indent |
| $\backslash shovecenter \quad \backslash framebox[5em]\{center\} \ \backslash$ | center |
| $\backslash shoveright \quad \backslash framebox[5em]\{right\}$                | right  |
| $\backslash >$   |        |

```

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

```

outdent

left

indent

right

---

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

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

---

```

\<=!
x+x \\\
x+x+x \\\
x+x+x+x \\\
x+x+x+x+x \\\
x+x+x+x+x+x \\\
x+x+x+x+x+x+x \\\
x+x+x+x+x+x+x+x \\\
\>

```

$x + x$ 
(47)

$x + x + x + x$ 
(48)

$x + x + x + x + x + x$ 
(49)

$x + x + x + x + x + x + x$ 
(50)

$x + x + x + x + x + x + x + x$ 
(51)

$x + x + x + x + x + x + x + x + x$ 
(52)

---

**mintagsep** (*key*) If the available space on a line does not suffice to place both the equation and its tag (with **\raisetag** a minimum separation of **mintagsep**; default is 0.5em), a tag will automatically be lowered or raised (depending on whether it is placed on the right or left). The macro **\raisetag** may be used to fine-tune the vertical placement (applies only if the tag is already shifted due to lack of space). For what it is worth, **\raisetag!** force-raises the tag and frees up the space occupied by the tag.

---

```

\[* \phi = -\int \frac{\mathrm{d}x}{\sqrt{1+x^2}} \]
\[* x = \frac{\partial}{\partial \phi} \sin \phi \]
\raisetag{0.45\baselineskip} \]

```

$$\phi = - \int \frac{dx}{\sqrt{1+x^2}}$$
(53)

$$x = \frac{\partial}{\partial \phi} \sin \phi$$
(54)

---

## 2.4 Punctuation

Extending proper punctuation across equations is a delicate matter, and maintaining it while redacting the text certainly takes more attention to detail than many authors are willing

to afford. A contributing factor is that punctuation marks are harder to spot alongside equation context and somewhat out of place anyway.

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

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

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

---

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

---

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

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

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

---

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

---

## 2.5 Math Classes at Alignment

Alignment in multi-line equations breaks equations into components before and after the alignment position. Unfortunately, this also interrupts 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

---

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



assumptions on the letters that will surround the alignment marker ‘&’ in order to obtain the appropriate spacing also across the alignment.

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

`class` (*key*) The package implements a more flexible assignment of math classes at the alignment. The `ampeq` (*key*) above default behaviour is invoked by the optional argument `class=ampeq` (or `ampeq` for short). The optional argument `class=eqamp` (or `eqamp` for short) imposes math classes at the alignment such that an equation sign should be placed just before the alignment. Concretely, it inserts `\mathrel{}` classes just before and after the alignment marker. Furthermore, in case of an empty left alignment cell, the leading math class is changed to `\mathord{}` so that a following binary operator is not interpreted as a unary one. For example, the following two expressions produce (almost) identical output:

---

|  |                             |
|--|-----------------------------|
| <pre>\&lt;[class=ampeq] a &amp;= b+c \\   &amp;= d-e \\   &amp;\mathrel{ }\phantom{=} +f \&gt;</pre> | $a = b + c$ $= d - e$ $+ f$ |
| <pre>\&lt;[class=eqamp] a =&amp; b+c \\   =&amp; d-e \\   &amp; +f \&gt;</pre>                       | $a = b + c$ $= d - e$ $+ f$ |

---

`classout` (*key*) Math classes just before and after alignment can be adjusted freely by the optional arguments:  
`classin` (*key*)  
`classlead` (*key*)

`classout={class},      classin={class},      classlead={class}.`

The parameter `classlead` alternatively `classin*` determines the math class just after the alignment if the cell before alignment is empty. The spacing at the alignment is determined by the pairing of the last/first character and the selected math class at the alignment:

|  |  |   |         |  |       |   |  |  |
|--|--|---|---------|--|-------|---|--|--|
|  |  | a | _a-out  |  | _in-b | b |  |  |
|  |  |   | _lead-c |  |       | c |  |  |

## 2.6 Vertical Spacing

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

The spacing is determined by combination of several aspects:

**Baselines.** First, T<sub>E</sub>X inserts some glue between lines of text to make them appear as regular as possible. The amount of inserted glue is determined by T<sub>E</sub>X's rules which depend on height, depth and intended baseline separation. This interline spacing also applies to the lines of displayed equations as well as the interfaces between text and displayed equations.

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

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

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

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

---

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

---

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

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

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

---

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

---

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

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

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

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

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

`par` (*key*) The key `par` controls whether the equation environments end in horizontal mode (value `cont`) or in vertical mode (value `par`, default) with a dedicated amount of glue `belowparskip`. An environment can also be made to end in vertical mode without interline skip (value `top`) using the glue `belowtopskip`.

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

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

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

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

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

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

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

## 2.7 Further Environments and Features

The package supplies some additional environments and features:

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

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

`colsep (key)` \_\_\_\_\_  
`frame (key)` `\[ \left\{`  
`wrap (key)` `\begin{equationsbox}[margin=1em]`  
`x &= \cos\phi \ \`  
`\phi &= \arccos x`  
`\end{equationsbox}`  
`\right\}\]`  
`\Longrightarrow\<=[shape=1,frame]`  
`x = \cos\phi &`  
`\phi = \arccos x \ \`  
`x = (z+z^{-1})/2 &`  
`\phi = -i\log z`  
`\>\Longleftarrow\`

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

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

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

---

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

---

|  |  |
|--|--|
| <pre>\eqnlineset   {subeqtemplate={#1-\roman{#2}}} \begin{subequations} \left[ x = \cos\phi \right] and \left[ \phi = \arccos x \right] \end{subequations}</pre> | $x = \cos \phi$ (55-i)<br>$\phi = \arccos x$ (55-ii) |
|--|--|

---

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

---

|  |                                       |
|--|---------------------------------------|
| <pre>\left[ x \&amp;= \cos\phi \intertext[medskip]{and} \phi \&amp;= \arccos x \right]</pre> | $x = \cos \phi$<br>$\phi = \arccos x$ |
|--|---------------------------------------|

---

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

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

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

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

---

`\eqncontrol{frametag[={cmd}]}`

---

|  |  |
|--|--|
| <pre>\left[ x \&amp;= \cos\phi \right] \framecell \phi \&amp;= \arccos x \right]  \left[ \framecell[\fboxrule2pt\fbox]   \mbox{important} \eqnpunct! \right]</pre> | $x = \cos \phi$<br><div style="border: 1px solid black; padding: 2px; display: inline-block;"><math>\phi = \arccos x</math></div><br><div style="border: 2px solid black; padding: 2px; display: inline-block;">important!</div> |
|--|--|

|  |  |   |
|--|--|---|
| <pre>\[! \framecell[\fcolorbox{white}{yellow}] \eqncontrol{frametag=\fbboxsep2pt\fbbox} \mbox{highlight}\]</pre> | <div style="background-color: yellow; padding: 2px 10px; display: inline-block;">highlight</div> | <div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">(56)</div> |
|--|--|---|

---

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

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

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

---

|  |                                       |
|--|---------------------------------------|
| <pre>\&lt;[alt={example equations}] x &amp;= \cos\phi \\ \eqnalt[line]{reverse relationship} \phi &amp;= \arccos x \&gt;</pre> | $x = \cos \phi$<br>$\phi = \arccos x$ |
|--|---------------------------------------|

---

## 2.8 General Options

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

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

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

The package supplies the following general settings:

| option                         | description  |
|--------------------------------|--|
| <code>defaults=classic</code>  | mimic classic L <sup>A</sup> T <sub>E</sub> X/amsmath (layout and dimensions)  |
| <code>defaults=eqnlines</code> | eqnlines layout with fontsize-relative dimensions  |
| <code>rescan</code>            | rescan environment body for special commands (e.g. <code>\verb</code> )  |
| <code>linesfallback</code>     | single column in align mode reverts to lines mode  |
|                                | value <code>reuse</code> avoids third measuring pass   |
| <code>ampproof</code>          | equip optional argument parsing with protection for ‘&’  |
| <code>crerror</code>           | invoke an error when ‘\’ is used in a single equation  |
| <code>modifierwarning</code>   | invoke a warning for unknown environment modifiers   |
| <code>scanpar</code>           | allow scanning of <code>\par</code> within equation body<br>(e.g., for use in nested <code>\parbox</code> or <code>minipage</code> ) |

## 2.9 Feature Selection and Package Options

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

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

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

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

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

| feature              | description  |
|----------------------|--|
| <i>env</i>           | provide/overwrite environment <i>env</i> :<br><code>equation</code> , <code>gather</code> , <code>multline</code> , <code>align</code> , <code>flalign</code> ,<br><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>\&lt;...\&gt;</code>   |
| <code>eqref</code>   | provide/overwrite macro <code>eqref</code>   |
| <code>tagform</code> | provide/overwrite macro <code>\tagform@</code>   |
| <code>maketag</code> | provide/overwrite macro <code>\maketag@@@</code>   |

## 3 Information

### 3.1 Copyright

Copyright © 2024–2025 Niklas Beisert

Based on the L<sup>A</sup>T<sub>E</sub>X package `amsmath`: Copyright © 1995, 2000, 2013 American Mathematical Society; 2016–2024 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X Project with contributions by the above and David Carlisle.

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

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

### 3.3 Files and Installation

The package consists of the files:

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

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

- Run (pdf)L<sup>A</sup>T<sub>E</sub>X on `eqnlines.dtx` to compile the manual `eqnlines.pdf` (this file).
- Run L<sup>A</sup>T<sub>E</sub>X on `eqnlines.ins` to create the package `eqnlines.sty` and the developers version `eqnlines-dev.sty`. Copy the file `eqnlines.sty` to an appropriate directory of your L<sup>A</sup>T<sub>E</sub>X distribution, e.g. `texmf-root/tex/latex/eqnlines`.

### 3.4 Related CTAN Packages

The package is related to other packages available at CTAN:

- This package uses the package `keyval` to process the options for the package, environments and macros. Compatibility with the `keyval` package has been tested with v1.15 (2022/05/29).



- This package reproduces the math environments functionality of the package `amsmath`. The present code is based on `amsmath` v2.17t (2024/11/05). Compatibility with the `amsmath` package is maintained whether `eqnlines` is loaded before or after `amsmath`. By default, `eqnlines` overwrites most math environments of `amsmath` with its own implementations. It can also preserve them as `ams...` if needed. Alternatively, `eqnlines` may assign individual names to the maths environments and preserve the ones of `amsmath`. The other features provided by `amsmath` can be used.
- The package `mathtools` is a popular extension of the `amsmath` package. This package incorporates some of the features and improvements provided by the `mathtools` package. Compatibility with the `mathtools` package has been tested with v1.31 (2024/10/04), and it is maintained whether `eqnlines` is loaded before or after `mathtools`. Some features like emphasising equations via `empheq` do not (yet) work.
- This package cooperates with the package `hyperref` to create anchors and references within the electronic document. Compatibility with the `hyperref` package has been tested with v7.01l (2024/11/05).
- This package supports the display of labels and references through the package `showkeys`. Compatibility with the `showkeys` package has been tested with v3.21 (2024/05/23).
- This package is currently not compatible with the package `cleveref` (thanks to Jonáš Dujava for pointing out). The command `\Cref` will not refer properly to equation numbers recorded by the `equations` environment. Further features of either package and/or/in combination with `amsmath` may fail due to the patching by the package. The alternative package `zref-clever` appears to work as intended. Incompatibility with the `cleveref` package has been observed for v0.21.4 (2018/03/27). Compatibility with the `zref-clever` package has been tested with v0.5.1 (2024/11/28).

### 3.5 Feature Suggestions

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

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

### 3.6 Revision History

**v0.9:** 2025/05/18

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

- fixed minor issues
- thanks to Jonáš Dujava for various reports and suggestions

**v0.8:** 2025/04/30

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

**v0.7.1:** 2025/04/09

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

**v0.7:** 2025/04/03

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

**v0.6.1:** 2025/03/27

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

**v0.6:** 2025/03/11

- preliminary PDF tagging support (<https://latex3.github.io/tagging-project/>; `amsmath` *must* be loaded *before* `eqnlines` to avoid errors)
- classic L<sup>A</sup>T<sub>E</sub>X/`amsmath` vs. `eqnlines` presets
- changed vertical spacing schemes and added further options
- supplied dimensions processed by `\glueexpr`
- more independent of `amsmath` structures
- internal reorganisations

**v0.5:** 2025/02/25

- preview version published on CTAN

## A Implementation

The appendix documents the various components of the present package.

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

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

## B General Support

In the following we describe general purpose supporting routines.

### B.1 Development Messages

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

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

### B.2 Supporting Definitions

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

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

**TODO:** describe

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

### B.3 Dollardollar Abstraction

`\dollar\dollar@begin` As of 2025 L<sup>A</sup>T<sub>E</sub>X defines `\dollar\dollar@begin` and `\dollar\dollar@end` to represent (and adjust) the beginning and end of bare T<sub>E</sub>X display equations (`'$$'`). For the time being, we make sure to revert to `'$$'` if these macros are not yet available:

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

### B.4 Look-Ahead in Alignment

Scanning for optional arguments [...] or modifiers such as `'*` using the L<sup>A</sup>T<sub>E</sub>X `\@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 macro invocation. For clarity, we copy L<sup>A</sup>T<sub>E</sub>X's original `\@ifnextchar` in `\kernel@ifnextchar` which skips over spaces as `\eql@ifnextchar@loose`. We replicate the `amsgen` version `\new@ifnextchar` that does not skip over spaces as `\eql@ifnextchar@loose`. The space before `#1` allows to look-ahead for spaces as well:

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

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

```

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

```

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

```

41 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
42 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
43 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
44 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
45 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{0}{#1}{%
46   \eq@ifnextgobble@loose\eq \@atxii{#1}{#2}}}
47 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{0}{#1}{%
48   \eq@ifnextgobble@tight\eq \@atxii{#1}{#2}}}
49 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[{#1}{#1[{#2}]}]{%
50   \long\def\eq@teststaropt@loose#1#2#3{%
51     \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}}
52   \long\def\eq@teststaropt@tight#1#2#3{%
53     \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}}
54   \long\def\eq@teststaroropt@loose#1#2#3{%
55     \eq@ifstar@loose{#1}{\eq@testopt@loose{#2}{#3}}}
56   \long\def\eq@teststaroropt@tight#1#2#3{%
57     \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}}
58   \long\def\eq@gobbleopt[#1]{}
59   \long\def\eq@gobbleoptone[#1]#2{}

```

**TODO:** describe

```

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

```

**TODO:** describe

```

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

```

```
80 \def\eql@parseopt@gobble#1{\eql@parseopt@peek}
```

`\eql@spbgroup` The second challenge is addressed by enclosing the look-ahead in spurious groups<sup>3</sup> which  
`\eql@spegroup` protect against triggering ‘&’. The macros `\eql@spbgroup` and `\eql@spegroup` open and  
`\eql@srbgroup` close a spurious group. For some reason, the look-ahead mechanism requires further  
`\eql@sregroup` protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end.  
 These adjustments are included in the macros `\eql@srbgroup` and `\ers@spegroup`:

```
81 \def\eql@spbgroup{\iffalse{\fi\ifnum0=‘}\fi}
82 \def\eql@spegroup{\ifnum0=‘{\fi\iffalse}\fi}
83 \def\eql@srbgroup{\relax\iffalse{\fi\ifnum0=‘}\fi}
84 \def\eql@sregroup{\let\@let@token\relax\ifnum0=‘{\fi\iffalse}\fi}
```

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

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

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

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

`\eql@amproof` We may want to replace L<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:

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

---

<sup>3</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>.

## B.5 Error Messages

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

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

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

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

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

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

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

## B.6 amsmath Integration

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

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

**TODO:** temporary fix for development stages

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

## B.7 PDF Tagging Support

`\eql@tagging@...` Proper PDF tagging<sup>4</sup> support requires a L<sup>A</sup>T<sub>E</sub>X (development) version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with L<sup>A</sup>T<sub>E</sub>X versions around 2020: **TODO:** adjust to further developments

```

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

```

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

```

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

```

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

```

181 \def\eql@tagging@start{%
182   \csname bool_if:N\expandafter\endcsname
183   \csname l__math_collected_bool\endcsname{%
184     \toks@{\expandafter{\eql@tagging@opt}}%
185     \edef\eql@tmp{\@currenvir}{\the\toks@} \the\eql@scan@reg}%
186     \csname __math_process:nn\expandafter\endcsname\eql@tmp

```

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



```

187     \@kernel@math@registered@begin
188     \csname bool_set_true:N\expandafter\endcsname
189     \csname l__math_collected_bool\endcsname
190   }%
191 }
192 \def\eql@tagging@end{}
193 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}
194 \else
195   \def\eql@tagging@mathsave{}
196   \def\eql@tagging@mathaddlast{}
197   \def\eql@tagging@tagbegin{}
198   \def\eql@tagging@tagend{}
199   \def\eql@tagging@tagsave{}
200   \def\eql@tagging@tagaddbox{}
201   \def\eql@tagging@tablesaveinner{}
202   \def\eql@tagging@tableaddinner{}
203   \def\eql@tagging@tablesavelines{}
204   \def\eql@tagging@tablesavealign{}
205   \def\eql@tagging@alignleft{}
206   \def\eql@tagging@aligncenter{}
207   \def\eql@tagging@alignright{}
208   \def\eql@tagging@start{}
209   \def\eql@tagging@end{}
210   \def\eql@tagging@register@env{\@gobble}
211 \fi

```

## B.8 Key-Value Processing

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

```

212 \RequirePackage{keyval}

```

### Value Selection.

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

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

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

```

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

```

```

224     \ifx\@tempa\@tempb
225     \edef\@tempd{\unexpanded\expandafter\expandafter\expandafter{%
226     \expandafter\@secondoftwo\@tempc}}}%
227     \fi
228     }%
229     \fi
230     }%
231     \ifdefined\@tempd
232     \@tempd
233     \else
234     \eqLError{undefined value '3' for option '2' of '1'}%
235     \fi
236 }

```

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

```

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

```

`\eqL@decide@if`

```

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

```

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

```

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

```

## Key Declaration.

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

```

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

```

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

```

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

```

## Options and Control Interface.

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

```

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

```

`\eqnaddopt`

```

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

```

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

```

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

```

`\eql@control@default`

```

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

```

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

```

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

```

## C Parameters and Registers

In the following, we collect parameter and register definitions.

## C.1 Parameters

**TODO:** describe

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

338 \def\eq@skip@long@above{\abovedisplayskip}
339 \def\eq@skip@long@below{\belowdisplayskip}
340 \def\eq@skip@short@above{\abovedisplaysshortskip}
341 \def\eq@skip@short@below{\belowdisplaysshortskip}
342 \def\eq@skip@cont@above{\eq@skip@short@above}
343 \def\eq@skip@cont@below{\eq@skip@short@below}
344 \def\eq@skip@par@above{\eq@skip@long@above}
345 \def\eq@skip@par@below{\eq@skip@long@below}
346 \def\eq@skip@top@above{\eq@skip@long@above}
347 \def\eq@skip@top@below{\eq@skip@long@below}
348 \def\eq@skip@med@above{\abovedisplayskip/2}
349 \def\eq@skip@med@below{\belowdisplayskip/2}
350 \def\eq@skip@tag@above{\z@skip}
351 \def\eq@skip@tag@below{\z@skip}
352 \def\eq@skip@partag@above{\z@skip}
353 \def\eq@skip@partag@below{\z@skip}
354 \def\eq@skip@medtag@above{\z@skip}
355 \def\eq@skip@medtag@below{\z@skip}

```

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

```

356 \newdimen\eq@colsepmin@
357 \def\eq@colsepmin@val{1em}
358 \def\eq@colsepmax@val{.5\maxdimen}

```

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

```

359 \newdimen\eq@tagwidthmin@
360 \eq@tagwidthmin@\z@

```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. T<sub>E</sub>X's built-in value is half a quad<sup>5</sup> in font number 2. As the tag is processed in text mode, we use 0.5em instead.

```

361 \newdimen\eq@tagsepmin@
362 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

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

```

\eq@equations@ang@opt
\eq@box@ang@opt
363 \def\eq@equations@sqr@opt{equation,nonumber}
364 \def\eq@equations@ang@opt{align,nonumber}
365 \def\eq@box@ang@opt{align}

```

## Multi-Line Align Mode.

```

366 \let\eq@columns@fulllength\eq@false

```

## C.2 Registers

**TODO:** describe

---

<sup>5</sup>another half of a quad is left empty at the other end of the line.

## General.

`\eql@cellbox@` (*box*) The box `\eql@cellbox@` holds the present alignment component and `\eql@tagbox@` the  
`\eql@tagbox@` (*box*) tag for the present line. The corresponding dimensions `\eql@cellwidth@` and  
`\eql@cellwidth@` (*dimen*) `\eql@tagwidth@` hold their widths. `\eql@prevwidth@` holds the width of the previous  
`\eql@prevwidth@` (*dimen*) alignment component: **TODO:** adjust  
`\eql@tagwidth@` (*dimen*)  
`\eql@prevdepth@` (*dimen*)  
`\eql@prevgraf@` (*counter*)  
367 `\newbox\eql@cellbox@`  
368 `\newbox\eql@tagbox@`  
369 `\newdimen\eql@cellwidth@`  
370 `\newdimen\eql@prevwidth@`  
371 `\newdimen\eql@tagwidth@`  
372 `\newdimen\eql@prevdepth@`  
373 `\newcount\eql@prevgraf@`  
  
`\eql@totalwidth@` (*dimen*)  
`\eql@tagwidth@max@` (*dimen*)  
374 `\newdimen\eql@totalwidth@`  
375 `\newdimen\eql@tagwidth@max@`  
  
`\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:  
  
376 `\newdimen\eql@line@height@`  
377 `\newdimen\eql@line@depth@`  
  
`\eql@line@width@` (*dimen*)  
`\eql@line@avail@` (*dimen*)  
`\eql@line@pos@` (*dimen*)  
`\eql@widthsep@` (*counter*)  
`\eql@availsep@` (*counter*)  
`\eql@possep@` (*counter*)  
`\eql@line@offset@` (*dimen*)  
378 `\newdimen\eql@line@width@`  
379 `\newdimen\eql@line@avail@`  
380 `\newdimen\eql@line@pos@`  
381 `\newcount\eql@line@availsep@`  
382 `\newcount\eql@line@widthsep@`  
383 `\newcount\eql@line@possep@`  
384 `\newdimen\eql@line@offset@`

## Rows and Columns.

`\eql@row@` (*counter*) `\eql@row@` counts the present row (1-based) and `\eql@totalrows@` holds the total number  
`\eql@totalrows@` (*counter*) of rows:  
`\eql@tagrows@` (*counter*)  
385 `\newcount\eql@row@`  
386 `\newcount\eql@totalrows@`  
387 `\newcount\eql@tagrows@`  
  
`\eql@column@`  
`\eql@totalcolumns@`  
388 `\newcount\eql@column@`  
389 `\newcount\eql@totalcolumns@`  
  
`\eql@colsep@` (*dimen*) The dimension of the intercolumn separation for align environments is stored in  
`\eql@colsep@`:  
390 `\newdimen\eql@colsep@`  
  
`\eql@intercolumns@` (*counter*)  
391 `\newcount\eql@intercolumns@`

## Vertical Spacing Adjustments.

`\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:

```
392 \newdimen\eqldisplay@firstavail@
393 \def\eqldisplay@firstavail@set#1{%
394   \ifnum\eqldisplay@row@=\@ne
395     \global\eqldisplay@firstavail@#1%
396   \fi
397 }
```

`\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:

```
398 \newcount\eqldisplay@raisefirstlast@
```

## Shared Registers.

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

```
399 \ifdefined\measuring@true\else
400   \expandafter\newif\csname ifmeasuring@\endcsname
401 \fi
402 \AddToHook{package/hyperref/after}{
403   \ifdefined\Hy@ifnotmeasuring
404     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
405   \fi
406 }
```

`\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:

```
407 \ifdefined\@displaytrue\else
408   \expandafter\newif\csname if@display\endcsname
409   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
410 \fi
```

## C.3 Hooks

`\eqldhook@...` For what it’s worth, we define a couple of entry points where one might hook into the equations typesetting framework. The  $\text{\LaTeX}$  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

```
411 \let\eqldhook@blockbefore\@empty
412 \let\eqldhook@blockafter\@empty
413 \let\eqldhook@blockin\@empty
414 \let\eqldhook@blockout\@empty
```



```

415 \let\eql@hook@linein\@empty
416 \let\eql@hook@lineout\@empty
417 \let\eql@hook@colin\@empty
418 \let\eql@hook@colout\@empty
419 \let\eql@hook@eqin\@empty
420 \let\eql@hook@eqout\@empty
421 \let\eql@hook@innerleft\@empty
422 \let\eql@hook@innerright\@empty
423 \let\eql@hook@innerlead\@empty

```

## D Features

### D.1 Punctuation

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

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

```

424 \let\eql@punct@col\@empty
425 \let\eql@punct@line\relax
426 \let\eql@punct@block\relax
427 \let\eql@punct@main\relax

```

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

```

428 \let\eql@punct@sep\relax

```

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

```

429 \def\eqnpunct{%
430   \eq@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
431 \def\eql@punct@next@set#1{%
432   \ifmmode
433     \def\eql@punct@col{#1}%
434     \def\eql@punct@line{#1}%
435     \def\eql@punct@block{#1}%
436   \else
437     \eqnaddopt{punct={#1}}%
438   \fi
439   \ignorespaces}
440 \def\eql@punct@next@setrelax{%
441   \ifmmode
442     \let\eql@punct@block\relax
443   \else
444     \eqnaddopt{punct*}%
445   \fi
446   \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:

```

447 \def\eql@punct@apply@col{%
448   \ifx\eql@punct@col\@empty\else
449     \eql@punct@sep
450     \eql@punct@col
451     \let\eql@punct@col\@empty
452   \fi
453 }

```

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`

```

454 \def\eql@punct@apply@line{%
455   \ifx\eql@punct@line\relax
456   % \TODO hand down immediately?
457   \else
458     \ifx\eql@punct@line\@empty\else
459       \eql@punct@sep
460       \eql@punct@line
461     \fi
462     \let\eql@punct@line\relax
463     \let\eql@punct@col\@empty
464   \fi
465 }

```

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

```

466 \def\eql@punct@apply@block{%
467   \ifx\eql@punct@block\relax
468   % \TODO hand down immediately?
469   \else
470     \ifx\eql@punct@block\@empty\else
471       \eql@punct@sep
472       \eql@punct@block
473     \fi
474     \let\eql@punct@block\relax
475     \let\eql@punct@line\relax
476     \let\eql@punct@col\@empty
477   \fi
478 }

479 \let\eqnpunctapply\eql@punct@apply@block

```

## D.2 Math Classes at Alignment

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

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

```

480 \def\eql@class@innerright@sel@{%
481   \ifdim\eql@prevwidth@=\z@
482     \eql@class@innerlead
483   \else
484     \eql@class@innerright

```

```

485 \fi
486 }

```

`@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`

```

487 \def\eq@class@innerleft@set#1{%
488   \def\eq@class@innerleft{#1}%
489 }
490 \def\eq@class@innerright@set#1{%
491   \def\eq@class@innerright{#1}%
492   \let\eq@class@innerright@sel\eq@class@innerright
493 }
494 \def\eq@class@innerlead@set#1{%
495   \def\eq@class@innerlead{#1}%
496   \let\eq@class@innerright@sel\eq@class@innerright@sel@
497 }

```

`\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):

```

498 \def\eq@class@ampeq{%
499   \eq@class@innerleft@set{}%
500   \eq@class@innerright@set{}}%
501 }
502 \def\eq@class@eqamp{%
503   \eq@class@innerleft@set{\mathrel{}}%
504   \eq@class@innerright@set{\mathrel{}}%
505   \eq@class@innerlead@set{}}%
506 }
507 \eq@class@ampeq

```

## D.3 Framed Cells

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

```

508 \let\eq@frame@cmd\@undefined
509 \newdimen\eq@frame@margin@
510 \def\eq@frame@set[#1]{%
511   \global\eq@append\eq@cell@container{\def\eq@frame@cmd{#1}}
512 \protected\def\framecell{\eq@testopt@tight@ampsafe\eq@frame@set\fbbox}
513 \def\eq@frame@measure{%
514   \setbox\z@\hbox{\eq@frame@cmd{}}%
515   \eq@frame@margin@.5\wd\z@
516 }
517 \def\eq@frame@print{%
518   \setbox\eq@cellbox@\hbox{%
519     \eq@frame@cmd{\unhbox\eq@cellbox@}%
520   }%
521 }
522 \def\eq@frame@adjust{%
523   \setbox\eq@cellbox@\hbox{%
524     \eq@frame@cmd{%
525       \unhbox\eq@cellbox@
526       \unkern
527       \unskip
528     }%
529     \hfil

```

```

530     \kern\z@
531   }%
532 }

```

## D.4 Alternative Content Description

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

```

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

```

## E Equation Numbering

**TODO:** describe

### E.1 Supporting Definitions

**Parameters.**

```

534 \let\eql@tags@autolabel\eql@false
535 \let\eql@tags@autotag\eql@true
536 \let\eql@tags@warn\eql@true

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

```

**Registers.**

```

538 \let\eql@numbering@mode\@undefined
539 \let\eql@numbering@measure@call\@empty

```

ring@target@ (*counter*)

```

540 \let\eql@numbering@active\eql@true
541 \newcount\eql@numbering@target@

542 \let\eql@tags@container\@undefined
543 \def\eql@tags@container@clear{%
544   \let\eql@tags@label\@undefined
545   \let\eql@tags@name\@undefined
546   \let\eql@tags@tag\@undefined
547   \let\eql@tags@ref\@undefined
548   \eql@raisetag@amount@\z@
549   \eql@raisetag@mode@\tw@
550 }

```

```

551 \let\eql@tags@label\@undefined
552 \let\eql@tags@name\@undefined

```

\eql@nexttag

```

553 \let\eql@tags@tag\@undefined
554 \let\eql@tags@ref\@undefined
555 \let\eql@tags@frame@cmd\@firstofone

```

setag@amount@ (*dimen*)

```
556 \newdimen\eq@raisetag@amount@
557 \newcount\eq@raisetag@mode@
```

tags@glabel@ (*counter*)

```
558 \newcount\eq@tags@glabel@
559 \eq@tags@glabel@z@
560 \def\eq@tags@glabel{equation.eql-\the\eq@tags@glabel@}
561 \def\eq@tags@glabel@step{\global\advance\eq@tags@glabel@\@ne}
```

## E.2 Schemes

**TODO:** describe

**Table.**

```
562 \def\eq@numbering@tab@first{first}
563 \def\eq@numbering@tab@last{last}
564 \def\eq@numbering@tab@middle{middle}
565 \def\eq@numbering@tab@here{here}
566 \def\eq@numbering@tab@best{best}
567 \def\eq@numbering@tab@in{in}
568 \def\eq@numbering@tab@out{out}
569 \def\eq@numbering@tab@sub{sub}
570 \def\eq@numbering@tab@all{all}

571 \def\eq@numbering@tab@off{off}
572 \def\eq@numbering@tab@on{on}
573 \def\eq@numbering@tab@single{single}
574 \def\eq@numbering@tab@multi{multi}
575 \def\eq@numbering@tab@none{none}
```

**TODO:** describe

```
576 \let\eq@numbering@tab@f\eq@numbering@tab@first
577 \let\eq@numbering@tab@l\eq@numbering@tab@last
578 \let\eq@numbering@tab@m\eq@numbering@tab@middle
579 \let\eq@numbering@tab@mid\eq@numbering@tab@middle
580 \let\eq@numbering@tab@o\eq@numbering@tab@out
581 \let\eq@numbering@tab@outside\eq@numbering@tab@out
582 \let\eq@numbering@tab@i\eq@numbering@tab@in
583 \let\eq@numbering@tab@inside\eq@numbering@tab@in
584 \let\eq@numbering@tab@within\eq@numbering@tab@in
585 \let\eq@numbering@tab@h\eq@numbering@tab@here
586 \let\eq@numbering@tab@center\eq@numbering@tab@middle
587 \let\eq@numbering@tab@c\eq@numbering@tab@middle
588 \let\eq@numbering@tab@optimal\eq@numbering@tab@best
589 \let\eq@numbering@tab@s\eq@numbering@tab@sub
590 \let\eq@numbering@tab@subeq\eq@numbering@tab@sub
591 \let\eq@numbering@tab@subequation\eq@numbering@tab@sub
592 \let\eq@numbering@tab@subequations\eq@numbering@tab@sub
593 \let\eq@numbering@tab@a\eq@numbering@tab@all
594 \let\eq@numbering@tab@n\eq@numbering@tab@none
595 \let\eq@numbering@tab@false\eq@numbering@tab@off
596 \let\eq@numbering@tab@true\eq@numbering@tab@on
597 \let\eq@numbering@tab@no\eq@numbering@tab@off
```

```

598 \let\eql@numbering@tab@yes\eql@numbering@tab@on
599 \let\eql@numbering@tab@0\eql@numbering@tab@multi
600 \eql@letcs{eql@numbering@tab@0}\eql@numbering@tab@off
601 \eql@letcs{eql@numbering@tab@!}\eql@numbering@tab@on
602 \eql@letcs{eql@numbering@tab@*}\eql@numbering@tab@off
603 \eql@letcs{eql@numbering@tab@1}\eql@numbering@tab@single
604 \eql@letcs{eql@numbering@tab@-}\eql@numbering@tab@none
605 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
606 \let\eql@numbering@mode\eql@numbering@tab@all
607 \let\eql@numbering@mode@multi\eql@numbering@tab@all
608 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

## Implementations.

```

609 \def\eql@numbering@init@all{\eql@numbering@target@\m@ne}
610 \def\eql@numbering@init@first{\eql@numbering@target@\@ne}
611 \def\eql@numbering@init@last{\eql@numbering@target@\@MM}
612 \def\eql@numbering@init@here{\eql@numbering@target@\z@}

```

**TODO:** describe

```

613 \def\eql@numbering@init@in{%
614   \ifdefined\eql@tagsleft
615     \eql@numbering@init@last
616   \else
617     \eql@numbering@init@first
618   \fi
619 }

```

**TODO:** describe

```

620 \def\eql@numbering@init@out{%
621   \ifdefined\eql@tagsleft
622     \eql@numbering@init@first
623   \else
624     \eql@numbering@init@last
625   \fi
626 }

```

**TODO:** describe

```

627 \def\eql@numbering@init@middle{%
628   \eql@numbering@target@\z@
629   \let\eql@numbering@measure@call\eql@numbering@measure@middle}
630 \def\eql@numbering@measure@middle{%
631   \ifnum\eql@numbering@target@=\z@
632     \eql@numbering@target@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
633   \fi
634 }

```

**TODO:** describe

```

635 \def\eql@numbering@init@best{%
636   \eql@numbering@target@\z@
637   \let\eql@numbering@measure@call\eql@numbering@measure@best
638 }
639 \def\eql@numbering@measure@best{%
640   \ifnum\eql@numbering@target@=\z@
641     \let\eql@numbering@best@use\eql@true
642     \eql@numbering@init@out
643   \fi

```

644 }

**TODO:** describe

```
645 \def\eq@numbering@init@sub{%
646   \eq@numbering@target@none
647   \ifdefined\eq@subequations@active\else
648     \let\eq@numbering@measure@call\eq@numbering@subeq@test
649     \let\eq@numbering@subeq@use\eq@true
650   \fi
651 }
```

**Selection.**

```
652 \def\eq@numbering@set#1{%
653   \ifcsname eq@numbering@tab@#1\endcsname
654     \let\eq@tmp\eq@numbering@mode
655     \expandafter\let\expandafter\eq@numbering@mode
656     \csname eq@numbering@tab@#1\endcsname
657     \ifx\eq@numbering@mode\eq@numbering@tab@off
658       \let\eq@numbering@mode\eq@tmp
659       \let\eq@numbering@active\eq@false
660     \fi
661     \ifx\eq@numbering@mode\eq@numbering@tab@on
662       \let\eq@numbering@mode\eq@tmp
663       \let\eq@numbering@active\eq@true
664     \fi
665     \ifx\eq@numbering@mode\eq@numbering@tab@none
666       \let\eq@numbering@mode\eq@numbering@mode@multi
667       \let\eq@numbering@active\eq@false
668     \fi
669     \ifx\eq@numbering@mode\eq@numbering@tab@single
670       \let\eq@numbering@mode\eq@numbering@mode@single
671       \let\eq@numbering@active\eq@true
672     \fi
673     \ifx\eq@numbering@mode\eq@numbering@tab@multi
674       \let\eq@numbering@mode\eq@numbering@mode@multi
675       \let\eq@numbering@active\eq@true
676     \fi
677     \ifx\eq@numbering@mode\eq@numbering@tab@all
678       \def\eq@tmp{multi}%
679     \else\ifx\eq@numbering@mode\eq@numbering@tab@sub
680       \def\eq@tmp{multi}%
681     \else
682       \def\eq@tmp{single}%
683     \fi\fi
684     \eq@letcs{eq@numbering@mode@\eq@tmp}\eq@numbering@mode
685   \else
686     \eq@error{numbering mode '#1' unknown: setting mode to 'all'}%
687     \let\eq@numbering@mode\eq@numbering@tab@all
688   \fi
689 }
```

**TODO:** describe

```
690 \def\eq@numbering@init{%
691   \let\eq@numbering@measure@call\@empty
692   \let\eq@numbering@subeq@use\eq@false
693   \csname eq@numbering@init@\eq@numbering@mode\endcsname
```

```

694 \ifdefined\eql@numbering@active
695   \let\eql@numbering@eqnswinit\@eqnswtrue
696 \else
697   \let\eql@numbering@eqnswinit\@eqnswfalse
698 \fi
699 \let\eql@numbering@active\eql@false
700 }

```

### E.3 Interface

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

```

701 \eql@amsmath@after{
702   \let\eql@print@eqnum@default\print@eqnum
703   \let\eql@incr@eqnum@default\incr@eqnum
704 }

```

**TODO:** describe

```

705 \protected\def\donumber{%
706   \if@eqnsw\else
707     \global\@eqnswtrue
708     \ifx\print@eqn\@empty
709       \global\let\print@eqn\eql@print@eqnum@default
710     \fi
711     \ifx\incr@eqn\@empty
712       \global\let\incr@eqn\eql@incr@eqnum@default
713     \fi
714   \fi
715 }

```

**TODO:** reconsider operation

`\numberhere`

```

716 \protected\def\eql@numberhere{%
717   \ifnum\eql@numbering@target@<\z@
718     \global\@eqnswtrue
719   \else
720     \global\eql@numbering@target@\eql@row@
721   \fi
722 }

```

**TODO:** describe

`\numbernext`

```

723 \protected\def\eql@numbernext{%
724   \ifnum\eql@numbering@target@<\z@
725     \global\@eqnswfalse
726   \else
727     \ifnum\eql@numbering@target@=\eql@row@
728       \global\advance\eql@numbering@target@\@one
729     \fi
730   \fi
731 }

```

**Activation Trigger.**



```

732 \def\eql@tags@autoenable{%
733   \global\@eqnswtrue
734   \ifx\eql@numbering@mode\eql@numbering@tab@here
735     \ifnum\eql@numbering@target@=\z@
736       \numberhere
737     \fi
738   \fi
739 }

```

**Labels.** **TODO:** describe

\eql@label@org

```

740 \let\eql@label@org\label

```

**TODO:** describe

```

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

```

**TODO:** describe

```

742 \protected\def\eql@label{%
743   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
744 }

```

**TODO:** describe

```

745 \def\eql@tags@add@labelname[#1]#2{%
746   \def\eql@tmp{#1}%
747   \ifx\eql@tmp\eql@testopt@default\else
748     \eql@tags@add@name{#1}%
749   \fi
750   \eql@tags@add@label{#2}%
751 }

```

**TODO:** describe

```

752 \def\eql@tags@set@label#1{%
753   \ifdefined\eql@tags@warn
754     \ifdefined\eql@tags@label
755       \eql@warn@label@multiple{#1}%
756     \fi
757   \fi
758   \def\eql@tags@label{#1}%
759 }

```

**TODO:** describe

```

760 \def\eql@tags@set@name#1{%
761   \ifdefined\eql@tags@warn
762     \ifdefined\eql@tags@name
763       \eql@warn@name@multiple
764     \fi
765   \fi
766   \def\eql@tags@name{#1}%
767 }

```

**TODO:** describe

```

768 \def\eql@tags@add@label#1{%
769   \ifdefined\eql@tags@autolabel
770     \eql@tags@autoenable

```

```

771 \fi
772 \global\eql@appendexpand\eql@tags@container{%
773     \noexpand\eql@tags@set@label{#1}}%
774 }

```

**TODO:** describe

```

775 \def\eql@tags@add@name#1{%
776     \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
777     \global\eql@appendmacro\eql@tags@container\eql@tmp
778 }

```

**TODO:** describe

```

779 \def\eql@tags@addblock@label#1{%
780     \eql@appendexpand\eql@tags@container@block{%
781         \noexpand\eql@tags@set@label{#1}}%
782 }

```

**TODO:** describe

```

783 \def\eql@tags@addblock@name#1{%
784     \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
785     \eql@appendmacro\eql@tags@container@block\eql@tmp
786 }

```

**Tags.** **TODO:** describe

\eql@tag@default

```

787 \protected\def\eql@tag@default{%
788     \eql@warn@here\tag
789     \eql@tag@gobble
790 }
791 \let\tag\eql@tag@default

```

\eql@tag@gobble

```

792 \def\eql@tag@gobble{%
793     \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}

```

**TODO:** describe

```

794 \protected\def\eql@tag{%
795     \eql@ampprotecttwo\eql@teststaropt@tight
796     {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
797     \eql@testopt@default
798 }

```

\eql@tags@add@tagref

```

799 \def\eql@tags@add@tagref[#1]#2{%
800     \def\eql@tmp{#1}%
801     \ifx\eql@tmp\eql@testopt@default\else
802         \eql@tags@add@ref{#1}%
803     \fi
804     \eql@tags@add@tag{#2}%
805 }

```

**TODO:** describe

```

806 \def\eql@tags@set@tag#1{%
807   \ifdefined\eql@tags@warn
808     \ifdefined\eql@tags@tag
809       \eql@warn@tag@multiple
810     \fi
811   \fi
812 \def\eql@tags@tag{#1}%
813 }

```

**TODO:** describe

```

814 \def\eql@tags@set@ref#1{%
815   \ifdefined\eql@tags@warn
816     \ifdefined\eql@tags@ref
817       \eql@warn@ref@multiple
818     \fi
819   \fi
820 \def\eql@tags@ref{#1}%
821 }

```

**TODO:** describe

```

822 \def\eql@tags@add@tag#1{%
823   \ifdefined\eql@tags@autotag
824     \eql@tags@autoenable
825   \fi
826 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
827 \global\eql@appendmacro\eql@tags@container\eql@tmp
828 }

```

**TODO:** describe

```

829 \def\eql@tags@add@ref#1{%
830   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
831   \global\eql@appendmacro\eql@tags@container\eql@tmp
832 }

```

tags@add@tagform@off

```

833 \def\eql@tags@add@tagform@off{%
834   \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
835 }

```

**TODO:** describe

```

836 \def\eql@tags@addblock@tag#1{%
837   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
838   \eql@appendmacro\eql@tags@container@block\eql@tmp
839 }

```

**TODO:** describe

```

840 \def\eql@tags@addblock@ref#1{%
841   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
842   \eql@appendmacro\eql@tags@container@block\eql@tmp
843 }

```

**TODO:** describe

```

844 \def\eql@tags@addblock@tagform@off{%
845   \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
846 }

```

## Raise Tags.

`\raisetag`

```
847 \def\eqL@raisetag@default{%  
848   \eqL@warn@here\raisetag  
849   \eqL@raisetag@gobble  
850 }  
  
851 \let\eqL@raisetag@gobble\@gobble
```

**TODO:** describe

```
852 \eqL@amsmath@let\raisetag\eqL@raisetag@default
```

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

```
853 \def\eqL@raisetag#1{%  
854   \def\@tempa{#1}%  
855   \def\@tempb{!}%  
856   \ifx\@tempa\@tempb  
857     \eqL@tags@add@raisemode\@ne  
858   \else  
859     \eqL@tags@add@raiseamount{#1}%  
860   \fi  
861 }  
  
862 \def\eqL@tags@add@raiseamount#1{%  
863   \global\eqL@appendexpand\eqL@tags@container{%  
864     \eqL@raisetag@amount@the\glueexpr#1\relax\relax}%  
865 }  
  
866 \def\eqL@tags@add@raisemode#1{%  
867   \global\eqL@append\eqL@tags@container{\eqL@raisetag@mode@#1}%  
868 }
```

## E.4 Integration

**TODO:** describe

**Support.** **TODO:** describe

```
869 \def\eqL@numbering@settools{%  
870   \let\label\eqL@label  
871   \let>tag\eqL@tag  
872   \let\raisetag\eqL@raisetag  
873   \let\numberhere\eqL@numberhere  
874   \let\numbernext\eqL@numbernext  
875 }
```

**TODO:** not necessary anymore

```
876 \def\eqL@numbering@settools@gobble{%  
877   \let\label\eqL@label@gobble  
878   \let>tag\eqL@tag@gobble  
879   \let\raisetag\eqL@raisetag@gobble  
880   \let\numberhere\relax  
881   \let\numbernext\relax  
882 }
```

```

883 \def\eql@numbering@autoblock{%
884   \begingroup
885     \let\eql@tags@warn\eql@false
886     \eql@tags@container@block
887     \ifdefined\eql@tags@autolabel
888       \ifdefined\eql@tags@label
889         \global\@eqnswtrue
890       \fi
891     \fi
892     \ifdefined\eql@tags@autotag
893       \ifdefined\eql@tags@tag
894         \global\@eqnswtrue
895       \fi
896     \fi
897   \endgroup
898 }

```

```

899 \def\eql@numbering@warnunused{%
900   \ifdefined\eql@tags@label
901     \eql@warn@label@unused
902   \fi
903   \ifdefined\eql@tags@name
904     \eql@warn@name@unused
905   \fi
906   \ifdefined\eql@tags@tag
907     \eql@warn@tag@unused
908   \fi
909   \ifdefined\eql@tags@erf
910     \eql@warn@ref@unused
911   \fi
912 }

```

**Single Line.** **TODO:** describe

```

913 \def\eql@numbering@single@init{%
914   \eql@numbering@target@>\m@ne
915   \eql@numbering@settools
916   \eql@numbering@eqnswinit
917   \eql@numbering@autoblock
918   \global\let\eql@tags@container\eql@tags@container@block
919   \let\eql@tags@warn\eql@true
920 }

```

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

```

921 \def\eql@numbering@measure@init{%
922   \eql@numbering@settools
923   \ifnum\eql@numbering@target@>\m@ne
924     \eql@numbering@eqnswinit
925     \eql@numbering@autoblock
926   \fi
927   \global\let\eql@tags@container\eql@tags@container@block
928   \let\eql@tags@warn\eql@true
929 }

```

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

```

930 \def\eql@numbering@measure@line@begin{%

```

```

931 \ifnum\eq\@numbering@target@<\z@
932   \global\eq\@numbering@eqnswinit
933 \fi
934 }

```

**TODO:** describe

```

935 \def\eq\@numbering@measure@eval{%
936   \let\eq\@numbering@best@use\eq\@false
937   \eq\@numbering@measure@call
938   \ifnum\eq\@numbering@target@>\eq\@totalrows@
939     \eq\@numbering@target@\eq\@totalrows@
940   \fi
941   \ifnum\eq\@numbering@target@>\z@
942     \if@eqnsw\else
943       \eq\@numbering@target@\z@
944     \fi
945   \fi
946 }

```

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

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

```

947 \def\eq\@numbering@print@init{%
948   \let\eq\@tags@warn\eq\@false
949   \ifnum\eq\@numbering@target@<\z@
950     \eq\@numbering@settools
951     \global\let\eq\@tags@container\eq\@tags@container@block
952   \else
953     \let\eq\@tags@container@block\eq\@tags@container
954     \eq\@numbering@settools@gobble
955   \fi
956 }

```

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

```

957 \def\eq\@numbering@print@block@begin{%
958   \ifnum\eq\@numbering@target@>\z@
959     \eq\@tags@makeblockanchor
960   \fi
961   \ifdefined\eq\@numbering@subeq@use
962     \eq\@tags@printsbeqlabel
963   \fi
964 }

```

**TODO:** describe

```

965 \def\eq\@numbering@print@line@begin{%
966   \ifnum\eq\@numbering@target@<\z@
967     \global\eq\@numbering@eqnswinit
968   \fi
969 }

```

**TODO:** describe

```

970 \def\eq\@numbering@print@line@eval{%
971   \ifnum\eq\@numbering@target@<\z@
972     \if@eqnsw
973       \eq\@tags@container

```

```

974   \fi
975 \else
976   \ifnum\eqL@numbering@target@=\eqL@row@
977     \eqnswtrue
978     \eqL@tags@container@block
979   \else
980     \eqnswfalse
981   \fi
982 \fi
983 }

```

## E.5 Component Display

**Showkeys Integration.** **TODO:** describe

```

984 \let\eqL@SK@loaded\eqL@false
985 \let\eqL@SK@label\@gobble
986 \let\eqL@SK@clearlabel\@empty
987 \let\eqL@SK@setlabel\@gobble
988 \let\eqL@SK@printlabel@right\@empty
989 \let\eqL@SK@printlabel@left\@empty
990 \let\eqL@SK@printlabel@line\@empty
991 \def\eqL@label@clean{\eqL@label@org}
992 \AddToHook{package/showkeys/after}{
993   \let\eqL@SK@loaded\eqL@true
994   \def\eqL@SK@label#1{\SK@SK@label#1}
995   \def\eqL@SK@clearlabel{\let\eqL@SK@lab\relax}
996   \eqL@SK@clearlabel
997   \def\eqL@SK@label#1>#2\SK@{%
998     \def\eqL@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
999   }
1000 \def\eqL@SK@setlabel#1{\SK@\eqL@SK@label#1}
1001 \def\eqL@SK@printlabel@right{%
1002   \ifx\eqL@SK@lab\relax\else
1003     \rlap{\kern\marginparsep\eqL@SK@lab}%
1004     \eqL@SK@clearlabel
1005   \fi
1006 }
1007 \def\eqL@SK@printlabel@left{%
1008   \ifx\eqL@SK@lab\relax\else
1009     \llap{\eqL@SK@lab\kern\marginparsep}%
1010     \eqL@SK@clearlabel
1011   \fi
1012 }
1013 \def\eqL@SK@printlabel@line{%
1014   \ifx\eqL@SK@lab\relax\else
1015     \dimen@ \prevdepth
1016     \nointerlineskip
1017     \ifdefined\eqL@tagsleft
1018       \llap{%
1019         \eqL@SK@lab
1020         \kern\marginparsep
1021       }%
1022     \eqL@SK@clearlabel
1023   \else
1024     \rlap{%
1025       \dimen@ \displaywidth
1026       \advance\dimen@ \marginparsep

```

```

1027         \kern\dimen@
1028         \eqL@SK@lab
1029     }%
1030     \fi
1031     \eqL@SK@clearlabel
1032     \prevdepth\dimen@
1033     \fi
1034 }
1035 \let\eqL@label@org\label
1036 \def\eqL@label@clean{\let\SK@\@gobbletwo\eqL@label@org}
1037 }

```

## Labels.

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

```

1038 \def\eqL@composetag@label{%
1039     \eqL@SK@clearlabel
1040     \ifdefined\eqL@tags@label
1041         \ifnum
1042             \ifnum\eqL@numbering@target@<\z@
1043                 \eqL@row@
1044             \else
1045                 \eqL@numbering@target@
1046             \fi=\eqL@row@
1047         \eqL@composetag@label@
1048     \fi
1049 \fi
1050 }

```

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

```

1051 \def\eqL@composetag@label@{%
1052     \eqL@SK@setlabel\eqL@tags@label
1053     \begingroup
1054         \ifnum\eqL@numbering@target@=\eqL@row@
1055             \eqL@tags@container@anchor
1056         \fi
1057         \ifdefined\eqL@tags@name
1058             \let\@currentlabelname\eqL@tags@name
1059         \else
1060             \let\@currentlabelname\eqL@tags@name@generic
1061         \fi
1062         \expandafter\eqL@label@clean\expandafter{\eqL@tags@label}%
1063     \endgroup
1064 }

```

**TODO:** describe

```

1065 \def\eqL@tags@printsubeqlabel{%
1066     \eqL@tags@container@parent
1067     \ifdefined\eqL@tags@label
1068         \eqL@tags@makeblockanchor
1069         \eqL@SK@setlabel\eqL@tags@label
1070         \begingroup
1071             \def\@currentcounter{equation}%
1072             \eqL@tags@container@anchor
1073             \let\@currentlabelname\eqL@tags@name@generic
1074             \protected@edef\@currentlabel{\p@equation\theparentequation}%

```



```

1075     \expandafter\eq\@label@clean\expandafter{\eq\@tags@label}%
1076     \endgroup
1077     \eq\@SK@printlabel@line
1078   \fi
1079 }

```

**Hyperref Anchors.** **TODO:** describe

```

1080 \let\eq\@Hy@anchor\@gobble
1081 \AddToHook{package/hyperref/after}{
1082   \def\eq\@Hy@anchor#1{%
1083     \Hy@raisedlink{\hyper@anchor{#1}}%
1084   }%
1085 }

```

**TODO:** describe

```

1086 \def\eq\@tags@makeblockanchor{%
1087   \eq\@tags@glabel@step
1088   \eq\@Hy@anchor\eq\@tags@glabel
1089   \global\edef\eq\@tags@container@anchor{%
1090     \def\noexpand\thepage{\thepage}%
1091     \def\noexpand\@currentHref{\eq\@tags@glabel}%
1092   }%
1093 }

```

**TODO:** describe

**eq\@composetag@anchor**

```

1094 \def\eq\@composetag@anchor{%
1095   \ifdefined\eq\@tags@tag
1096     \def\@currentcounter{equation}%
1097     \ifdefined\eq\@tags@ref
1098       \let\@currentlabel\eq\@tags@ref
1099     \else
1100       \protected@edef\@currentlabel{\p@equation\eq\@tags@tag}%
1101     \fi
1102     \eq\@tags@glabel@step
1103     \edef\@currentHref{\eq\@tags@glabel}%
1104     \eq\@Hy@anchor\@currentHref
1105   \else
1106     \refstepcounter{equation}%
1107     \protected@edef\eq\@tags@tag{\theequation}%
1108   \fi
1109 }

```

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

```

1110 \def\eq\@tags@taglayout@set@direct#1{%
1111   \def\eq\@tags@taglayout##1{#1}%
1112 }
1113 \def\eq\@tags@taglayout@set#1{%
1114   \def\eq\@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1115 }

```

**TODO:** describe

```

1116 \def\eq\@tags@tagform@set@direct#1{%

```

```

1117 \def\eql@tags@tagform##1{#1}%
1118 }
1119 \def\eql@tags@tagform@set#1#2#3{%
1120 \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1121 }

1122 \eql@tags@taglayout@set{#1}
1123 \eql@tags@tagform@set({#1})
1124 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1125 \protected\def\tagform{\eql@tags@tagform}
1126 \protected\def\tagbox{\eql@tags@taglayout}
1127 \protected\def\tagboxed{\eql@tags@tagcompose}

```

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

```

1128 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1129 \def\eql@composetag@tag{%
1130 \eql@tagging@tagbegin
1131 \eql@tags@frame@cmd{%
1132 \eql@tags@taglayout{%
1133 \eql@tags@tagform\eql@tags@tag
1134 \eql@tagging@tagsave
1135 }%
1136 }%
1137 \eql@tagging@tagend
1138 }

```

## E.6 Tag Composition

**TODO:** describe

```

1139 \def\eql@composetag@measure{%
1140 \ifdefined\eql@tags@tag\else
1141 \stepcounter{equation}%
1142 \let\eql@tags@tag\theequation
1143 \fi
1144 \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1145 \ifnum\eql@numbering@target@<\z@
1146 \global\let\eql@tags@container\eql@tags@container@clear
1147 \fi
1148 }

```

**TODO:** describe

```

1149 \def\eql@composetag@print{%
1150 \eql@composetag@anchor
1151 \eql@composetag@label
1152 \ifdefined\eql@tags@left
1153 \eql@SK@printlabel@left
1154 \eql@composetag@tag
1155 \else
1156 \eql@composetag@tag
1157 \eql@SK@printlabel@right
1158 \fi

```

```

1159 \global\let\eql@tags@container\eql@tags@container@clear
1160 }

```

**TODO:** describe

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

```

1161 \def\eql@tagbox@make#1{%
1162   \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
1163   \eql@tagwidth@\wd\eql@tagbox@
1164   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1165     \eql@tagwidth@\eql@tagwidthmin@
1166   \fi
1167   \advance\eql@tagwidth@\eql@tagsepmin@
1168 }

```

**TODO:** describe

```

1169 \def\eql@tagbox@print@tagsright{%
1170   \ifnum\eql@raisetag@mode@=\tw@
1171     \kern-\wd\eql@tagbox@
1172     \box\eql@tagbox@
1173   \else
1174     \eql@tagbox@print@tagsright@raise
1175   \fi
1176 }

```

**TODO:** describe

```

1177 \def\eql@tagbox@print@tagsleft{%
1178   \ifnum\eql@raisetag@mode@=\tw@
1179     \wd\eql@tagbox@\z@
1180     \box\eql@tagbox@
1181   \else
1182     \eql@tagbox@print@tagsleft@raise
1183   \fi
1184 }

```

**TODO:** describe

```

1185 \def\eql@tagbox@print@tagsright@raise{%
1186   \ifnum\eql@row@=\eql@totalrows@
1187     \global\advance\eql@display@raisefirstlast@\tw@
1188   \fi
1189   \llap{\vtop{%
1190     \vskip-\eql@raisetag@amount@
1191     \normalbaselines
1192     \setbox\@ne\hbox{}}%
1193     \dp\@ne\eql@line@depth@
1194     \box\@ne
1195     \box\eql@tagbox@
1196   }}%
1197 }

```

**TODO:** describe

```

1198 \def\eql@tagbox@print@tagsleft@raise{%
1199   \ifnum\eql@row@=\@ne
1200     \global\advance\eql@display@raisefirstlast@\@ne
1201   \fi
1202   \rlap{\vbox{%
1203     \normalbaselines

```

```

1204 \box\eq\tagbox@
1205 \vbox to\eq\line@height@{}%
1206 \vskip\eq\raisetag@amount@
1207 }}%
1208 }

```

eq\tagbox@print@cell

```

1209 \def\eq\tagbox@print@cell{%
1210 \ifdefined\eq\tagleft
1211 \ifnum\eq\raisetag@mode@>\z@
1212 \ifdim\eq\tagwidth@>\dimexpr\eq\line@avail@+\eq\tagfuzz@\relax
1213 \eq\raisetag@mode@\@ne
1214 \fi
1215 \eq\display@firstavail@set\z@
1216 \eq\tagbox@print@tagleft
1217 \fi
1218 \kern\displaywidth
1219 \else
1220 \kern\displaywidth
1221 \ifnum\eq\raisetag@mode@>\z@
1222 \ifdim\eq\tagwidth@>
1223 \dimexpr\displaywidth-\eq\line@width@+\eq\tagfuzz@\relax
1224 \eq\raisetag@mode@\@ne
1225 \fi
1226 \eq\tagbox@print@tagright
1227 \fi
1228 \fi
1229 }

```

## F Subequation Numbering

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

### F.1 Definitions

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

```

1230 \eq\amsmath@undefine\c@parentequation
1231 \eq\amsmath@undefine\theparentequation
1232 \ifdefined\c@parentequation\else
1233 \newcounter{parentequation}
1234 \fi

```

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

```

1235 \def\eq\subequations@template{\theparentequation\alph{equation}}

```

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

```

1236 \let\eq\subequations@active\eq>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`:

```

1237 \def\eql@subequations@init{%
1238   \edef\eql@subequations@restorecounter{%
1239     \global\c@equation\the\c@equation\relax}%
1240   \eql@tags@container@block
1241   \ifdefined\eql@tags@tag
1242     \eql@tags@glabel@step
1243     \protected@edef\theHparentequation{\eql@tags@glabel}%
1244     \protected@edef\theparentequation{\eql@tags@tag}%
1245   \else
1246     \advance\c@equation\@ne
1247     \protected@edef\theparentequation{\theequation}%
1248     \ifdefined\theHequation
1249       \protected@edef\theHparentequation{\theHequation}%
1250     \fi
1251   \fi
1252   \global\c@parentequation\c@equation
1253   \global\c@equation\z@
1254   \let\theequation\eql@subequations@template
1255   \def\theHequation{\theHparentequation.\arabic{equation}}%
1256 }
```

`\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:

```

1257 \def\eql@subequations@close{%
1258   \ifnum\c@equation=\z@
1259     \eql@subequations@restorecounter
1260   \else
1261     \global\c@equation\c@parentequation
1262   \fi
1263 }
```

## F.2 Environment

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

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

```

1264 \def\eql@subequations@start{%
1265   \let\eql@tags@container@block\eql@tags@container@clear
1266   \eql@nextopt@process{subequations}%
1267   \eql@subequations@init
1268   \eql@tags@glabel@step
1269   \edef\eql@subequations@currentHref{\eql@tags@glabel}%
1270   \eql@Hy@anchor\eql@subequations@currentHref
1271   \edef\eql@subequations@thepage{\thepage}%
1272   \def\@currentcounter{equation}%
1273   \let\@currentHref\eql@subequations@currentHref
```

```

1274 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1275 \eq\@tags@container@block
1276 \ifdefined\eq\@tags@name
1277   \let\@currentlabelname\eq\@tags@name
1278 \else
1279   \let\@currentlabelname\eq\@tags@name@generic
1280 \fi
1281 \let\eq\@subequations@active\eq\@true
1282 \ifdefined\eq\@tags@label
1283   \eq\@SK@label\eq\@tags@label
1284 \fi
1285 \ignorespaces
1286 }

```

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

```

1287 \def\eq\@subequations@end{%
1288   \ifnum\c@equation>\z@
1289     \eq\@tags@container@block
1290     \ifdefined\eq\@tags@label
1291       \begingroup
1292         \def\@currentcounter{equation}%
1293         \let\thepage\eq\@subequations\thepage
1294         \let\@currentHref\eq\@subequations\@currentHref
1295 % \TODO how about tag* ?! also within equations!
1296         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1297         \ifdefined\eq\@tags@name
1298           \let\@currentlabelname\eq\@tags@name
1299         \else
1300           \let\@currentlabelname\eq\@tags@name@generic
1301         \fi
1302         \expandafter\eq\@label@clean\expandafter{\eq\@tags@label}%
1303       \endgroup
1304     \fi
1305   \fi
1306   \eq\@subequations@close
1307   \ignorespacesafterend
1308 }

```

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

```

1309 \newenvironment{eq\@subequations}{%
1310 (dev)\eq\@dev@enterenv
1311 \eq\@subequations@testall\eq\@subequations@start%
1312 }{%
1313 \eq\@subequations@end
1314 (dev)\eq\@dev@leaveenv
1315 }

```

**TODO:** describe

```

1316 \def\eq\@subequations@testall{\eq\@parseopt\eq\@subequations@parseopt}
1317 \def\eq\@subequations@parseopt{%
1318   \ifx\eq\@parseopt@token[%]
1319     \let\eq\@parseopt@next\eq\@parseopt@opt
1320   \fi
1321   \ifx\eq\@parseopt@token\eq\@atxi

```

```

1322 \let\eql@parseopt@next\eql@parseopt@label
1323 \fi
1324 \ifx\eql@parseopt@token\eql@atxii
1325 \let\eql@parseopt@next\eql@parseopt@label
1326 \fi
1327 \ifx\eql@parseopt@token\label
1328 \let\eql@parseopt@next\eql@parseopt@end
1329 \fi
1330 }

```

### F.3 Subequation Scheme

**TODO:** describe

```

1331 \def\eql@numbering@subeq@init{%
1332 \let\eql@save@theequation\theequation
1333 \let\eql@save@theHequation\theHequation
1334 \eql@subequations@init
1335 \let\eql@tags@container@parent\eql@tags@container@block
1336 \let\eql@tags@container@block\eql@tags@container@clear
1337 }

```

**TODO:** describe

```

1338 \def\eql@numbering@subeq@test{%
1339 \ifnum\c@equation<\tw@
1340 \let\eql@numbering@subeq@use\@ne
1341 \fi
1342 }

```

**TODO:** describe

```

1343 \def\eql@numbering@subeq@revert{%
1344 \let\eql@tags@container@block\eql@tags@container@parent
1345 \let\eql@numbering@subeq@use\eql@false
1346 \let\theequation\eql@save@theequation
1347 \let\theHequation\eql@save@theHequation
1348 \eql@subequations@restorecounter
1349 }

```

**TODO:** describe

```

1350 % \TODO note must not use setcounter here (when calc is loaded)
1351 \def\eql@numbering@subeq@close{%
1352 \eql@subequations@close
1353 }

```

## G Display Equations Support

**TODO:** describe

```

1354 \let\eql@interline@container\@undefined
1355 \def\eql@interline@container@clear{%
1356 \eql@displaybreak@open@\@MM
1357 \eql@vspaceskip@\z@skip
1358 \let\eql@vspaceskip@fixed\eql@false
1359 }

```

## G.1 Display Breaks

**TODO:** describe

erdisplaylinepenalty

```
1360 \interdisplaylinepenalty\@M
```

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

```
1361 \def\eq@getdsp@pen#1{%
1362   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1363 }
```

**TODO:** allow a displaybreak before equations

```
1364 \protected\def\eq@displaybreak@default{%
1365   \eq@warning{Invalid use of \string\displaybreak}{}%
1366   \eq@teststaroropt@loose\@gobble\eq@gobbleopt{}}
1367 \eq@amsmath@after{\let\eq@displaybreak@default\displaybreak}
1368 \eq@amsmath@let\displaybreak\eq@displaybreak@default

1369 \newcount\eq@displaybreak@pen@
1370 \newcount\eq@displaybreak@prepen@
1371 \newcount\eq@displaybreak@postpen@
```

**TODO:** describe

```
1372 \protected\def\eq@displaybreak{%
1373   \relax
1374   \eq@ampprotecttwo\eq@teststaroropt@tight
1375   \eq@displaybreak@star\eq@displaybreak@level{4}%
1376 }

1377 \def\eq@displaybreak@star#1{%
1378   \global\eq@appendexpand\eq@interline@container{%
1379     \eq@displaybreak@pen@\the\numexpr#1\relax\relax}%
1380 }

1381 \def\eq@displaybreak@level[#1]{%
1382   \ifnum#1<\z@
1383     \global\eq@append\eq@interline@container{\eq@displaybreak@pen@\@MM}%
1384   \else
1385     \global\eq@appendexpand\eq@interline@container{%
1386       \eq@displaybreak@pen@-\@getpen{#1}\relax}%
1387   \fi
1388 }
```

**TODO:** describe

```
1389 \def\eq@displaybreak@pre#1{%
1390   \ifnum#1<\z@
1391     \eq@displaybreak@prepen@\@MM
1392   \else
1393     \eq@displaybreak@prepen@-\@getpen{#1}\relax
1394   \fi
1395 }
```

**TODO:** describe

```
1396 \def\eq@displaybreak@post#1{%
```



```

1397 \ifnum#1<\z@
1398   \eqldisplaybreak@postpen@MM
1399 \else
1400   \eqldisplaybreak@postpen@-\getpen{#1}\relax
1401 \fi
1402 }

```

**TODO:** describe

```

1403 \def\eqldisplaybreak@inter#1{%
1404   \ifnum#1<\z@
1405     \interdisplaylinepenaltyM
1406   \else
1407     \interdisplaylinepenalty\eqldgetdspopen{#1}\relax
1408   \fi
1409 }

```

## G.2 Explicit Vertical Space

**TODO:** describe

`\eqlvspaceskip@` (*skip*)

```

1410 \newskip\eqlvspaceskip@
1411 \let\eqlvspaceskip@fixed\eqldfalse

1412 \let\eqlvspace@org\vspace
1413 \def\eqlvspace{%
1414   \ifvmode
1415     \expandafter\eqlvspace@immediate
1416   \else
1417     \expandafter\eqlvspace@line
1418   \fi
1419 }

1420 \def\eqlvspace@line{%
1421   \eqldifstar@loose{\eqlvspace@setfixed\eqlvspace@set}\eqlvspace@set
1422 }
1423 \def\eqlvspace@setfixed{%
1424   \global\eqldappend\eqldinterline@container{%
1425     \let\eqlvspaceskip@fixed\eqldtrue}%
1426 }
1427 \def\eqlvspace@set#1{%
1428   \global\eqldappendexpand\eqldinterline@container{%
1429     \advance\eqlvspaceskip@the\glueexpr#1\relax\relax}%
1430 }

1431 \def\eqlvspace@immediate{%
1432   \noalign\bgroup
1433     \eqldifstar@loose\eqlvspace@fixed\eqlvspace@discordable
1434 }
1435 \def\eqlvspace@fixed#1{%
1436   \dimen@prevdepth
1437   \hrule\@height\z@
1438   \penaltyM
1439   \vskip\glueexpr#1\relax
1440   \prevdepth\dimen@
1441 \egroup
1442 }

```

```

1443 \def\eql@vspace@discardable#1{%
1444     \skip\glueexpr#1\relax
1445     \egroup
1446 }

```

## G.3 Default Vertical Spacing

**TODO:** describe

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

```

1447 \newbox\eql@strutbox@
1448 \def\eql@strut@depth{.3}
1449 \def\eql@strut{\copy\eql@strutbox@}
1450 \let\eql@strut@cell\eql@strut
1451 \let\eql@strut@tag\eql@strut
1452 \def\eql@strut@make{%
1453     \setbox\eql@strutbox@\hbox{%
1454         \@tempdima\dimexpr
1455             \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1456         \@tempdima\dimexpr
1457             \normalbaselineskip-\normallineskiplimit-\@tempdima\relax
1458         \vrule\@height\@tempdima\@depth\@tempdima\@width\z@
1459     }
1460 }
1461 \AtBeginDocument{\eql@strut@make}

```

**TODO:** describe

```

1462 \def\eql@spread@set{%
1463     \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
1464     +\normalbaselineskip-\baselineskip\relax
1465     \ifdim\eql@spread@>\z@
1466         \openup\eql@spread@
1467         \ifdefined\spread@equation
1468             \let\spread@equation\@empty
1469         \fi
1470     \fi
1471 }

```

## G.4 Entry and Exit

**TODO:** describe

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

```

1472 \newskip\eql@abovespace@
1473 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

1474 \def\eql@display@enter{%
1475     \ifvmode
1476         \eql@prevdepth@\prevdepth
1477         \nointerlineskip
1478         \noindent

```

```

1479 \else
1480   \eql@prevdepth@\maxdimen
1481 \fi
1482 }

```

`\eql@display@adjust`

```

1483 \def\eql@display@adjust{%
1484   \ifdefined\eql@display@linewidth
1485     \displaywidth\glueexpr\eql@display@linewidth\relax
1486     \advance\displaywidth-\displayindent
1487   \fi
1488   \ifdefined\eql@display@marginleft
1489     \advance\displaywidth\displayindent
1490     \displayindent\glueexpr\eql@display@marginleft\relax
1491     \advance\displaywidth-\displayindent
1492   \fi
1493   \ifdefined\eql@display@marginright
1494     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1495   \fi
1496   \ifdim\displaywidth<\z@
1497     \displaywidth\z@
1498   \fi
1499 }

```

`\eql@display@init`

```

1500 \def\eql@display@init{%
1501   \let\displaybreak\eql@displaybreak
1502   \let\eql@vspace@org\vspace
1503   \let\vspace\eql@vspace
1504   \let\eqncontrol\eql@control
1505   \eql@spread@set
1506   \eql@strut@make
1507   \let\eql@frame@cmd\@undefined
1508 }

```

`\eql@display@print`

```

1509 \def\eql@display@print{%
1510   \eql@display@firstavail@\z@
1511   \eql@display@raisefirstlast@\z@
1512   \global\let\eql@interline@container\eql@interline@container@clear
1513 }

```

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

```

1514 \def\eql@display@halign@init#1{%
1515   \eql@row@\z@
1516   \eql@prevgraf@\prevgraf
1517   \everycr{\noalign{%
1518     \global\advance\eql@row@\@ne
1519     \prevgraf\numexpr\prevgraf+\@ne\relax
1520     #1%
1521   }}%
1522 }

```

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

```

1523 \def\eqldisplay@halign@start{%
1524   \prevgraf\numexpr\eql@prevgraf@+\@ne\relax
1525   \ifdim\eql@prevdepth@=\maxdimen\else
1526     \prevdepth\eql@prevdepth@
1527   \fi
1528   \ifdim\prevdepth=-\@m\p@\else
1529     \ifdefined\eqldisplay@height
1530       \skip@\baselineskip
1531       \advance\skip@-\glueexpr\eqldisplay@height\relax
1532       \advance\skip@-\prevdepth\relax
1533       \ifdim\skip@<\lineskiplimit
1534         \skip@\lineskip
1535       \fi
1536       \advance\skip@-\eql@spread@\relax
1537       \vskip\skip@
1538       \nointerlineskip
1539     \else
1540       \vskip-\eql@spread@\relax
1541     \fi
1542   \fi
1543 }

```

**TODO:** describe

```

1544 \def\eqldisplay@vspace{%
1545   \advance\abovedisplayskip\eql@abovespace@
1546   \advance\belowdisplayskip\eql@belowspace@
1547   \advance\belowdisplayskip\eql@vspaceskip@
1548 }

```

**TODO:** describe **TODO:** implement fixed \eql@vspaceskip@fixed

```

1549 \def\eqldisplay@vspace@native{%
1550   \advance\abovedisplayskip\eql@abovespace@
1551   \advance\belowdisplayskip\eql@belowspace@
1552   \advance\belowdisplayskip\eql@vspaceskip@
1553   \advance\abovedisplayshortskip\eql@abovespace@
1554   \advance\belowdisplayshortskip\eql@belowspace@
1555   \advance\belowdisplayshortskip\eql@vspaceskip@
1556 }

```

**TODO:** describe

```

1557 \def\eqldisplay@penalty{%
1558   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1559     \postdisplaypenalty\eqldisplaybreak@postpen@
1560   \fi
1561   \ifnum\eqldisplaybreak@open@=\@MM\else
1562     \postdisplaypenalty\eqldisplaybreak@open@
1563   \fi
1564   \ifnum\eqldisplaybreak@prepen@=\@MM\else
1565     \predisplaypenalty\eqldisplaybreak@prepen@
1566   \fi
1567 }

```

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

```

1568 \def\eqldisplay@halign@end{%
1569   \eql@interline@container
1570   \ifdefined\eql@vspaceskip@fixed
1571     \penalty\@M

```

```

1572 \vskip\eql@vspaceskip@
1573 \eql@vspaceskip@z@skip
1574 \global\eql@append\eql@interline@container{\eql@vspaceskip@z@skip}%
1575 \fi
1576 \global\eql@prevgraf@\numexpr\prevgraf+ \@ne\relax
1577 \ifdefined\eql@display@depth
1578 \prevdepth\glueexpr\eql@display@depth\relax
1579 \fi
1580 }

```

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

```

1581 \def\eql@display@close{%
1582 % \TODO temporary fix for development stages
1583 \ifdefined\eql@tagging@on
1584 \ifdefined\dollardollar@begin\else
1585 \belowdisplayskip-\belowdisplayskip
1586 \belowdisplayshortskip-\belowdisplayshortskip
1587 \fi
1588 \fi
1589 \ifdim\eql@display@firstavail@<\z@
1590 \eql@display@firstavail@z@
1591 \fi
1592 \eql@skip@mode@leave@z@
1593 \ifdim\eql@prevdepth@=\maxdimen
1594 \ifdim\predisplaysize=-\maxdimen
1595 \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1596 \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1597 \else
1598 \eql@skip@mode@above@z@
1599 \eql@skip@mode@below@z@
1600 \advance\eql@display@firstavail@\displayindent
1601 \ifdim\eql@display@firstavail@>\predisplaysize
1602 \ifcase\eql@skip@mode@short\relax
1603 \or
1604 \eql@skip@mode@above@\@ne
1605 \or
1606 \eql@skip@mode@above@\@ne
1607 \ifnum\eql@totalrows@=\@ne
1608 \eql@skip@mode@below@\@ne
1609 \fi
1610 \or
1611 \eql@skip@mode@above@\@ne
1612 \eql@skip@mode@below@\@ne
1613 \fi
1614 \fi
1615 \fi
1616 \else
1617 \ifdim\eql@prevdepth@=-\@m\p@
1618 \eql@skip@mode@above@\eql@skip@mode@top@above\relax
1619 \eql@skip@mode@below@\eql@skip@mode@top@below\relax
1620 \else
1621 \eql@skip@mode@above@\eql@skip@mode@par@above\relax
1622 \eql@skip@mode@below@\eql@skip@mode@par@below\relax
1623 \fi
1624 \fi
1625 \ifcase\eql@skip@mode@above@

```

```

1626 \or\or\or
1627 \predisplaypenalty\z@
1628 \or
1629 \predisplaypenalty\z@
1630 \fi
1631 \ifcase\eq\@skip@mode@below@
1632 \or\or\or
1633 \eq\@skip@mode@leave@\@ne
1634 \or
1635 \eq\@skip@mode@leave@\tw@
1636 \fi
1637 \ifdefined\eq\@skip@force@above
1638 \eq\@skip@mode@above@\eq\@skip@force@above\relax
1639 \fi
1640 \ifdefined\eq\@skip@force@below
1641 \eq\@skip@mode@below@\eq\@skip@force@below\relax
1642 \fi
1643 \ifdefined\eq\@skip@force@leave
1644 \eq\@skip@mode@leave@\eq\@skip@force@leave\relax
1645 \fi
1646 \ifnum\eq\@skip@mode@leave@>\z@
1647 \postdisplaypenalty\z@
1648 \fi
1649 \ifodd\eq\@display@raisefirstlast@
1650 \ifcase\eq\@skip@mode@above@
1651 \abovedisplayskip\glueexpr\eq\@skip@tag@above\relax
1652 \or
1653 \abovedisplayskip\glueexpr\eq\@skip@tag@above\relax
1654 \or
1655 \abovedisplayskip\glueexpr\eq\@skip@tag@above\relax
1656 \or
1657 \abovedisplayskip\glueexpr\eq\@skip@partag@above\relax
1658 \or
1659 \abovedisplayskip\glueexpr\eq\@skip@partag@above\relax
1660 \or
1661 \abovedisplayskip\z@skip
1662 \or
1663 \abovedisplayskip\glueexpr\eq\@skip@medtag@above\relax
1664 \or
1665 \abovedisplayskip\glueexpr\eq\@skip@custom@above\relax
1666 \fi
1667 \else
1668 \ifcase\eq\@skip@mode@above@
1669 \abovedisplayskip\glueexpr\eq\@skip@long@above\relax
1670 \or
1671 \abovedisplayskip\glueexpr\eq\@skip@short@above\relax
1672 \or
1673 \abovedisplayskip\glueexpr\eq\@skip@cont@above\relax
1674 \or
1675 \abovedisplayskip\glueexpr\eq\@skip@par@above\relax
1676 \or
1677 \abovedisplayskip\glueexpr\eq\@skip@top@above\relax
1678 \or
1679 \abovedisplayskip\z@skip
1680 \or
1681 \abovedisplayskip\glueexpr\eq\@skip@med@above\relax
1682 \or
1683 \abovedisplayskip\glueexpr\eq\@skip@custom@above\relax

```

```

1684 \fi
1685 \fi
1686 \ifnum\eqldisplay@raisefirstlast@>\@ne
1687 \ifcase\eqlskip@mode@below@
1688 \belowdisplayskip\glueexpr\eqlskip@tag@below\relax
1689 \or
1690 \belowdisplayskip\glueexpr\eqlskip@tag@below\relax
1691 \or
1692 \belowdisplayskip\glueexpr\eqlskip@tag@below\relax
1693 \or
1694 \belowdisplayskip\glueexpr\eqlskip@partag@below\relax
1695 \or
1696 \belowdisplayskip\glueexpr\eqlskip@partag@below\relax
1697 \or
1698 \belowdisplayskip\z@skip
1699 \or
1700 \belowdisplayskip\glueexpr\eqlskip@medtag@below\relax
1701 \or
1702 \belowdisplayskip\glueexpr\eqlskip@custom@below\relax
1703 \fi
1704 \else
1705 \ifcase\eqlskip@mode@below@
1706 \belowdisplayskip\glueexpr\eqlskip@long@below\relax
1707 \or
1708 \belowdisplayskip\glueexpr\eqlskip@short@below\relax
1709 \or
1710 \belowdisplayskip\glueexpr\eqlskip@cont@below\relax
1711 \or
1712 \belowdisplayskip\glueexpr\eqlskip@par@below\relax
1713 \or
1714 \belowdisplayskip\glueexpr\eqlskip@top@below\relax
1715 \or
1716 \belowdisplayskip\z@skip
1717 \or
1718 \belowdisplayskip\glueexpr\eqlskip@med@below\relax
1719 \or
1720 \belowdisplayskip\glueexpr\eqlskip@custom@below\relax
1721 \fi
1722 \fi
1723 \global\eqlskip@mode@leave@\eqlskip@mode@leave@
1724 \eql@interline@container
1725 \eqldisplay@penalty
1726 \eqldisplay@vspace
1727 % \TODO temporary fix for development stages
1728 \ifdefined\eql@tagging@on
1729 \ifdefined\dollardollar@begin\else
1730 \belowdisplayskip-\belowdisplayskip
1731 \fi
1732 \fi
1733 }

```

**TODO:** describe

```

1734 \def\eqldisplay@leave{%
1735 \prevgraf\eql@prevgraf@
1736 \ifcase\eqlskip@mode@leave@
1737 \or
1738 \endgraf
1739 \or

```

```

1740 \endgraf
1741 \prevdepth-\@m\p@
1742 \fi
1743 }

```

**TODO:** describe

```

1744 \def\eqldisplay@nest{%
1745 \let\label\eqldlabel@org
1746 \let\tag\eqldtag@default
1747 \let\raisetag\eqldraisetag@default
1748 \let\displaybreak\eqldisplaybreak@default
1749 \let\intertext\eqldintertext@default
1750 \let\vspace\eqldvspace@org
1751 }

```

**TODO:** describe

```

1752 \eqldappend\@arrayparboxrestore{%
1753 \eqlddisplay@nest
1754 \ifdefined\eqldampproof@active
1755 \eqldamprevert
1756 \fi
1757 \displayfalse
1758 }

```

## G.5 Stack

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

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

```

1759 \def\eqldstack@enable{%
1760 \let\eqldstack@save@equations\eqldstack@save@equations@
1761 \let\eqldstack@save@box\eqldstack@save@box@
1762 }

```

**TODO:** describe

```

1763 \let\eqldstack@save@equations\eqldstack@enable
1764 \let\eqldstack@save@box\eqldstack@enable
1765 \let\eqldstack@restore\@empty

```

**TODO:** describe

```

1766 \def\eqldstack@save@reg#1{\global#1\the#1\relax}
1767 \def\eqldstack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

**TODO:** describe

```

1768 \def\eqldstack@save@equations@{%
1769 \let\eqldstack@numbering@eqnswinit\eqldnumbering@eqnswinit
1770 \let\eqldstack@cell@container\eqldcell@container
1771 \let\eqldstack@tags@container\eqldtags@container
1772 \let\eqldstack@tags@container@anchor\eqldtags@container@anchor
1773 \let\eqldstack@interline@container\eqldinterline@container
1774 \let\eqldstack@widthdata@tab\eqldwidthdata@tab
1775 \edef\eqldstack@restore{%
1776 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
1777 \eqldstack@save@let\eqldstack@numbering@eqnswinit\eqldnumbering@eqnswinit

```



```

1778 \eqL@stack@save@let\eqL@stack@cell@container\eqL@cell@container
1779 \eqL@stack@save@let\eqL@stack@tags@container\eqL@tags@container
1780 \eqL@stack@save@let
1781 \eqL@stack@tags@container@anchor\eqL@tags@container@anchor
1782 \eqL@stack@save@let\eqL@stack@interline@container\eqL@interline@container
1783 \eqL@stack@save@let\eqL@stack@widthdata@tab\eqL@widthdata@tab
1784 \eqL@stack@save@reg\eqL@display@firstavail@
1785 \eqL@stack@save@reg\eqL@display@raisefirstlast@
1786 \eqL@stack@save@reg\eqL@column@
1787 \eqL@stack@save@reg\eqL@totalcolumns@
1788 \eqL@stack@save@reg\eqL@line@avail@
1789 \eqL@stack@save@reg\eqL@line@pos@
1790 \eqL@stack@save@reg\eqL@line@width@
1791 \eqL@stack@save@reg\eqL@line@depth@
1792 \eqL@stack@save@reg\eqL@line@height@
1793 \eqL@stack@save@reg\eqL@tagwidth@max@
1794 \eqL@stack@save@reg\eqL@numbering@target@
1795 \eqL@stack@save@reg\eqL@row@
1796 \eqL@stack@save@reg\eqL@tagrows@
1797 }%
1798 }

```

**TODO:** describe

```

1799 \def\eqL@stack@save@box@{%
1800 \let\eqL@stack@cell@container\eqL@cell@container
1801 \edef\eqL@stack@restore{%
1802 \eqL@stack@save@let\eqL@stack@cell@container\eqL@cell@container
1803 \eqL@stack@save@reg\eqL@row@
1804 }%
1805 }

```

## H Multi-Line Support

**TODO:** describe

### H.1 Measure Support

**TODO:** describe

```

1806 \def\eqL@measure@init#1#2{%
1807 \eqL@widthdata@reset
1808 \eqL@numbering@measure@init
1809 \eqL@row@z@
1810 \tabskip\z@skip
1811 \everycr{\noalign{%
1812 \global\advance\eqL@row@\@ne
1813 #1%
1814 }}%
1815 \global\let\eqL@interline@container\eqL@interline@container@clear
1816 \eqL@measure@savestate
1817 \measuring@true
1818 \eqL@display@halign@letcr{#2}%
1819 }

```

**TODO:** describe

```

1820 \def\eql@measure@tag{%
1821   \eql@tagwidth@ \z@
1822   \ifnum\eql@numbering@target@<\z@
1823     \if@eqnsw
1824       \eql@tags@container
1825       \eql@tagbox@make\eql@composetag@measure
1826       \ifnum\eql@raisetag@mode@=\@ne
1827         \eql@tagwidth@ \z@
1828       \fi
1829     \fi
1830   \fi
1831   \eql@widthdata@savetag
1832 }

```

**TODO:** describe

```

1833 \def\eql@measure@close{%
1834   \advance\eql@row@-\tw@
1835   \eql@totalrows@\eql@row@
1836   \eql@numbering@measure@eval
1837   \begingroup
1838     \eql@tags@container
1839     \ifnum\eql@numbering@target@>\z@
1840       \eql@tagbox@make\eql@composetag@measure
1841       \ifnum\eql@raisetag@mode@=\@ne
1842         \eql@tagwidth@ \z@
1843       \fi
1844       \eql@widthdata@savetagsingle
1845     \else
1846       \eql@numbering@warnunused
1847     \fi
1848   \endgroup
1849   \eql@measure@restorestate
1850 }

```

measure@restorestate

ql@measure@savestate

```

1851 \let\eql@measure@restorestate\@empty
1852 \def\eql@measure@savestate{%
1853   \begingroup
1854     \def\@elt##1{%
1855       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
1856     \global\edef\@gtempa{%
1857       \cl@@ckpt
1858       \let\noexpand\eql@measure@restorestate\noexpand\@empty
1859       \ifmeasuring@\noexpand\measuring@true\else\noexpand\measuring@false\fi
1860     }%
1861   \endgroup
1862   \let\eql@measure@restorestate\@gtempa
1863 }

```

## H.2 Line Breaks

**TODO:** describe

\eql@display@cr

```

1864 \protected\def\eql@display@cr{%

```

```

1865 \eql@ampprotecttwo\eql@teststaropt@tight{%
1866   \global\eql@append\eql@interline@container{\eql@displaybreak@pen@\@M}%
1867   \eql@display@cr@opt}
1868   \eql@display@cr@opt\z@skip
1869 }

```

\eql@display@cr@opt

```

1870 \def\eql@display@cr@opt[#1]{%
1871   \eql@display@endline
1872   \cr

1873   \noalign{%
1874     \eql@interline@container
1875     \ifdefined\eql@vspaceskip@fixed
1876       \penalty\@M
1877       \vskip\eql@vspaceskip@
1878       \eql@vspaceskip@\z@skip
1879     \fi
1880     \ifnum\eql@displaybreak@pen@=\@MM
1881       \penalty\interdisplaylinepenalty
1882     \else
1883       \penalty\eql@displaybreak@pen@
1884     \fi
1885     \advance\eql@vspaceskip@\glueexpr#1\relax
1886     \vskip\eql@vspaceskip@
1887     \global\let\eql@interline@container\eql@interline@container@clear
1888   }%
1889 }

```

display@halign@letcr

```

1890 \def\eql@display@halign@letcr#1{%
1891   \let\\\eql@display@cr
1892   \let\eql@display@endline#1%
1893 }

```

### H.3 Intertext

**TODO:** describe

**TODO:** revert in everymath?

```

1894 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
1895 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

1896 \def\eql@intertext@process{%
1897   \eql@display@endline
1898   \cr
1899   \ifmeasuring@
1900     \expandafter\@gobble
1901   \else
1902     \expandafter\eql@intertext@print
1903   \fi
1904 }

```

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

```

1905 \def\eq@intertext@print#1{%
1906   \noalign{%
1907     \eq@display@halign@end
1908     \let\eq@skip@force@below\z@
1909     \let\eq@skip@force@above\z@
1910     \eq@setkeys{intertext}\eq@intertext@opt
1911     \openup-\eq@spread@
1912     \penalty\postdisplaypenalty
1913     \ifcase\eq@skip@force@below\relax
1914       \advance\eq@vspaceskip@\glueexpr\eq@skip@long@below\relax
1915     \or
1916       \advance\eq@vspaceskip@\glueexpr\eq@skip@short@below\relax
1917     \or
1918       \advance\eq@vspaceskip@\glueexpr\eq@skip@cont@below\relax
1919     \or
1920       \advance\eq@vspaceskip@\glueexpr\eq@skip@par@below\relax
1921     \or
1922       \advance\eq@vspaceskip@\glueexpr\eq@skip@top@below\relax
1923     \or
1924       \advance\eq@vspaceskip@\z@skip
1925     \or
1926       \advance\eq@vspaceskip@\glueexpr\eq@skip@med@below\relax
1927     \or
1928       \advance\eq@vspaceskip@\glueexpr\eq@skip@custom@below\relax
1929     \fi
1930     \vskip\eq@vspaceskip@
1931     \global\let\eq@interline@container\eq@interline@container@clear
1932     \vbox{%
1933       \@parboxrestore
1934       \ifdim
1935         \ifdim@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
1936       \else
1937         \parshape\@ne
1938         \@totalleftmargin\linewidth
1939       \fi
1940       \noindent
1941       \prevgraf\eq@prevgraf@
1942       \ignorespaces
1943       #1%
1944       \par
1945       \global\eq@prevgraf@\prevgraf
1946     }%
1947     \penalty\predisplaypenalty
1948     \ifcase\eq@skip@force@above\relax
1949       \vskip\glueexpr\eq@skip@long@above\relax
1950     \or
1951       \vskip\glueexpr\eq@skip@short@above\relax
1952     \or
1953       \vskip\glueexpr\eq@skip@cont@above\relax
1954     \or
1955       \vskip\glueexpr\eq@skip@par@above\relax
1956     \or
1957       \vskip\glueexpr\eq@skip@top@above\relax
1958     \or
1959       \vskip\z@skip
1960     \or
1961       \vskip\glueexpr\eq@skip@med@above\relax
1962     \or

```

```

1963     \vskip\glueexpr\eq\@skip@custom@above\relax
1964     \fi
1965 %     \eq\@prevdepth@\maxdimen
1966     \eq\@prevdepth@\z@
1967     \eq\@display@\halign@start
1968 }
1969 }

```

**TODO:** describe

```

1970 \newenvironment{eq\@intertext}{%
1971   \eq\@testopt@tight\eq\@intertext@{}%
1972 }{%
1973   \aftergroup\eq\@intertext@after
1974   \ignorespacesafterend
1975 }

```

**TODO:** describe

```

1976 \def\eq\@intertext@env{intertext}
1977 \def\eq\@intertext@[#1]{%
1978   \global\def\eq\@intertext@opt{#1}%
1979   \ifx\@currenvir\eq\@intertext@env
1980     \expandafter\eq\@scan@env\expandafter\eq\@intertext@inject
1981   \else
1982     \expandafter\eq\@intertext@process
1983   \fi
1984 }

```

**TODO:** describe

```

1985 \def\eq\@intertext@inject{%
1986   \global\edef\eq\@intertext@after{%
1987     \noexpand\eq\@intertext@process{%
1988       \ifx\eq\@scan@body\eq\@scan@body@dump
1989         \eq\@scan@body@dump
1990       \else
1991         \noexpand\scantokens{\eq\@scan@body@dump}%
1992       \fi
1993     }%
1994   }%
1995 }

```

## I Column Placement

**TODO:** describe

### I.1 Supporting Definitions

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

```

1996 \newcount\eq\@shape@pos@
1997 \newdimen\eq\@shape@amount@
1998 \let\eq\@shape@lastrow\eq\@false

```

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

`\eql@marginleft@min@ (dimen)`

`\eql@marginright@ (dimen)`

`\eql@centeroffset@ (dimen)`

```

1999 \newdimen\eql@marginleft@
2000 \newdimen\eql@marginright@
2001 \newdimen\eql@marginleft@min@
2002 \newdimen\eql@centeroffset@

```

## I.2 Shape Schemes

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

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

```

2003 \def\eql@shape@tab@default{default}
2004 \def\eql@shape@tab@left{left}
2005 \def\eql@shape@tab@center{center}
2006 \def\eql@shape@tab@right{right}
2007 \def\eql@shape@tab@first{first}
2008 \def\eql@shape@tab@hanging{hanging}
2009 \def\eql@shape@tab@steps{steps}

```

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

```

2010 \let\eql@shape@tab@def\eql@shape@tab@default
2011 \let\eql@shape@tab@\eql@shape@tab@default
2012 \let\eql@shape@tab@l\eql@shape@tab@left
2013 \let\eql@shape@tab@c\eql@shape@tab@center
2014 \let\eql@shape@tab@r\eql@shape@tab@right
2015 \let\eql@shape@tab@rc\eql@shape@tab@first
2016 \let\eql@shape@tab@indent\eql@shape@tab@first
2017 \let\eql@shape@tab@hang\eql@shape@tab@hanging
2018 \let\eql@shape@tab@lcr\eql@shape@tab@hanging
2019 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
2020 \let\eql@shape@tab@lcr\eql@shape@tab@steps

```

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

```

2021 \let\eql@shape@mode\eql@shape@tab@default

```

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

```

2022 \def\eql@shape@set#1{%
2023   \ifcsname eql@shape@tab@#1\endcsname
2024     \expandafter\let\expandafter\eql@shape@mode
2025       \csname eql@shape@tab@#1\endcsname
2026   \else
2027     \eq@error{shape '#1' unknown: setting to default}%
2028     \let\eql@shape@mode\eql@shape@tab@default
2029   \fi
2030 }

```

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

```

2031 \def\eql@shape@layoutcenter@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2032 \def\eql@shape@layoutcenter@center{\eql@shape@pos@ne@\eql@shape@amount@z@}
2033 \def\eql@shape@layoutcenter@right{\eql@shape@pos@tw@\eql@shape@amount@z@}

```

```

2034 \let\eqL@shape@layoutcenter@default\eqL@shape@layoutcenter@center
2035 \def\eqL@shape@layoutleft@left{\eqL@shape@pos@z@z\eqL@shape@amount@z@}
2036 \def\eqL@shape@layoutleft@center{\eqL@shape@pos@ne\eqL@shape@amount@z@}
2037 \def\eqL@shape@layoutleft@right{\eqL@shape@pos@tw@z\eqL@shape@amount@z@}
2038 \let\eqL@shape@layoutleft@default\eqL@shape@layoutleft@left

```

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

```

2039 \def\eqL@shape@layoutcenter@first{%
2040   \eqL@shape@pos@z@
2041   \eqL@shape@amount@z@
2042   \ifnum\eqL@totalrows@>\@ne
2043     \ifnum\eqL@row@=\@ne
2044       \eqL@shape@amount@\eqL@indent@
2045     \fi
2046   \fi
2047 }
2048 \def\eqL@shape@layoutleft@first{%
2049   \eqL@shape@pos@z@
2050   \eqL@shape@amount@z@
2051   \ifnum\eqL@totalrows@>\@ne
2052     \ifnum\eqL@row@=\@ne
2053       \eqL@shape@amount@\eqL@indent@
2054     \fi
2055   \fi
2056 }

```

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

```

2057 \def\eqL@shape@layoutcenter@hanging{%
2058   \eqL@shape@pos@z@
2059   \eqL@shape@amount@\eqL@indent@
2060   \ifnum\eqL@totalrows@>\@ne
2061     \ifnum\eqL@row@=\@ne
2062       \eqL@shape@amount@z@
2063     \fi
2064   \fi
2065 }
2066 \def\eqL@shape@layoutleft@hanging{%
2067   \eqL@shape@pos@z@
2068   \eqL@shape@amount@z@
2069   \ifnum\eqL@totalrows@>\@ne
2070     \ifnum\eqL@row@=\@ne
2071       \eqL@shape@amount@-\eqL@indent@
2072     \fi
2073   \fi
2074 }

```

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

```

2075 \def\eqL@shape@layoutcenter@steps{%
2076   \eqL@shape@amount@z@
2077   \eqL@shape@pos@ne
2078   \ifnum\eqL@totalrows@>\@ne
2079     \ifnum\eqL@row@=\@ne

```

```

2080     \eql@shape@pos@\z@
2081   \fi
2082   \ifnum\eql@row@=\eql@totalrows@
2083     \eql@shape@pos@\tw@
2084   \fi
2085 \fi
2086 }
2087 \def\eql@shape@layoutleft@steps{%
2088   \eql@shape@pos@\z@
2089   \eql@shape@amount@\z@
2090   \ifnum\eql@totalrows@>\@ne
2091     \ifnum\eql@row@=\@ne
2092       \eql@shape@amount@-\eql@indent@
2093     \fi
2094     \ifnum\eql@row@=\eql@totalrows@
2095       \eql@shape@amount@\eql@indent@
2096     \fi
2097   \fi
2098 }

```

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

```

2099 \let\eql@shape@eval\undefined
2100 \def\eql@shape@select{%
2101   \expandafter\let\expandafter\eql@shape@eval
2102   \csname eql@shape%
2103     @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
2104     @\eql@shape@mode\endcsname
2105 }

```

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

```

2106 \protected\def\eql@shape@alignleft{%
2107   \global\eql@append\eql@cell@container{\eql@shape@pos@\z@}%
2108   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2109 \protected\def\eql@shape@aligncenter{%
2110   \global\eql@append\eql@cell@container{\eql@shape@pos@\@ne}%
2111   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2112 \protected\def\eql@shape@alignright{%
2113   \global\eql@append\eql@cell@container{\eql@shape@pos@\tw@}%
2114   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2115 \def\eql@shape@align@testpar#1{%
2116   \eql@ifstar@tight{#1[\eql@indent@]}%
2117   {\eql@ifnextgobble@tight{!}{#1[-\eql@indent@]}}%
2118   {\eql@testopt@tight{#1}\z@}}
2119 \def\eql@shape@alignamount@opt[#1]{\eql@shape@alignamount@set{#1}}

```

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

```

2120 \protected\def\eql@shape@alignamount{%
2121   \eql@ampprotecttwo\eql@ifstar@tight
2122   \eql@shape@alignamount@set\eql@shape@alignamount@add}
2123 \def\eql@shape@alignamount@add#1{%
2124   \global\eql@appendexpand\eql@cell@container{%
2125     \advance\eql@shape@amount@\the\glueexpr#1\relax\relax}}
2126 \def\eql@shape@alignamount@set#1{%
2127   \global\eql@appendexpand\eql@cell@container{%

```



```

2128 \eq@shape@amount@the\glueexpr#1\relax\relax}}
2129 \def\eq@shape@align@enable{%
2130 \let\shoveleft\eq@shape@alignleft
2131 \let\shovecenter\eq@shape@aligncenter
2132 \let\shoveright\eq@shape@alignright
2133 \let\shoveby\eq@shape@alignamount
2134 }

```

**TODO:** describe

```

2135 \protected\def\eq@shape@align@default{%
2136 \eq@warn@here{\shove...}%
2137 \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2138 \protected\def\eq@shape@align@amount@default{%
2139 \eq@warn@here{\shove...}%
2140 \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2141 \def\eq@shape@align@disable{%
2142 \let\shoveleft\eq@shape@align@default
2143 \let\shovecenter\eq@shape@align@default
2144 \let\shoveright\eq@shape@align@default
2145 \let\shoveby\eq@shape@align@amount@default
2146 }

```

### I.3 Width Data

`\width@single@` (*dimen*)

```

2147 \newdimen\eq@tagwidth@single@

```

`\eq@widthdata@tab` **TODO:** new

```

2148 \let\eq@widthdata@tab\@empty

```

`\eq@widthdata@reset`

```

2149 \def\eq@widthdata@reset{%
2150 \let\eq@widthdata@tab\@empty
2151 \eq@tagwidth@max@z@
2152 \eq@tagrows@z@
2153 }

```

`\eq@widthdata@add`

```

2154 \def\eq@widthdata@add#1{%
2155 \global\eq@appendexpand\eq@widthdata@tab{#1}%
2156 }

```

`\widthdata@startrow`

```

2157 \def\eq@widthdata@startrow{%
2158 \eq@widthdata@add{\eq@row@the\eq@row@relax}%
2159 }

```

`\widthdata@savecell`

```

2160 \def\eq@widthdata@savecell{%
2161 \eq@widthdata@add{%
2162 \eq@shape@pos@the\eq@shape@pos@relax
2163 \eq@cellwidth@the\eq@cellwidth@relax

```

```

2164 \eq\shape@amount@\the\eq\shape@amount@\relax
2165 \noexpand\eq\widthdata@cellcall
2166 }%
2167 }

```

ql@widthdata@savesep

```

2168 \def\eq\widthdata@savesep{%
2169 \eq\widthdata@add{\noexpand\eq\widthdata@sepcall}%
2170 }

```

ql@widthdata@savetag

```

2171 \def\eq\widthdata@savetag{%
2172 \eq\widthdata@add{\eq\tagwidth@\the\eq\tagwidth@\relax;}%
2173 \ifdim\eq\tagwidth@>\eq\tagwidth@max@
2174 \global\eq\tagwidth@max@\eq\tagwidth@
2175 \fi
2176 \ifdim\eq\tagwidth@>\z@
2177 \global\advance\eq\tagrows@\@ne
2178 \fi
2179 }

```

thdata@savetagsingle

```

2180 \def\eq\widthdata@savetagsingle{%
2181 \global\eq\tagwidth@single@\eq\tagwidth@
2182 \global\eq\tagwidth@max@\eq\tagwidth@
2183 \global\eq\tagrows@\@ne
2184 }

```

\eq\widthdata@for

```

2185 \def\eq\widthdata@for#1{%
2186 \def\eq\widthdata@forall{#1}%
2187 \expandafter\eq\widthdata@forstep\eq\widthdata@tab
2188 \eq\row@0\relax\eq\tagwidth@\z@\relax;%
2189 }

```

ql@widthdata@forstep

```

2190 \def\eq\widthdata@forstep\eq\row@#1\relax#2\eq\tagwidth@#3\relax;%
2191 \eq\row@#1\relax
2192 \ifnum\eq\row@=\z@\else
2193 \eq\tagwidth@#3\relax
2194 \def\eq\widthdata@cells{#2}%
2195 \eq\widthdata@forall
2196 \expandafter\eq\widthdata@forstep
2197 \fi
2198 }

```

\eq\widthdata@get

```

2199 \def\eq\widthdata@get#1{%
2200 \eq\row@#1\relax
2201 \expandafter\eq\widthdata@getdef\expandafter{\the\eq\row@}%
2202 \expandafter\eq\widthdata@getparse\eq\widthdata@tab\@nil%
2203 }

```

`\eql@widthdata@getdef`

```
2204 \def\eql@widthdata@getdef#1{%
2205   \def\eql@widthdata@getparse
2206     ##1\eql@row@#1\relax##2\eql@tagwidth@##3\relax;##4\@nil{%
2207     \eql@tagwidth@##3\relax
2208     \def\eql@widthdata@cells{##2}%
2209   }%
2210 }
```

`\eql@colwidth@tab`

```
2211 \let\eql@colwidth@tab\@empty
```

`\eql@colwidth@get`

```
2212 \def\eql@colwidth@get#1{%
2213   \ifcase\expandafter#1\eql@colwidth@tab\else\z@ \fi
2214 }
```

`\eql@colwidth@save`

```
2215 \def\eql@colwidth@save#1{%
2216   \edef\eql@colwidth@tab{%
2217     \noexpand\or\the#1%
2218     \unexpanded\expandafter{\eql@colwidth@tab}%
2219   }%
2220 }
```

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

```
2221 \def\eql@widthdata@calc{%
2222   \eql@column@\z@
2223   \eql@line@pos@\z@
2224   \eql@line@possep@\z@
2225   \eql@line@avail@\eql@totalwidth@
2226   \eql@line@availsep@\eql@intercolumns@
2227   \eql@line@width@\z@
2228   \eql@line@widthsep@\z@
2229   \let\eql@widthdata@cellcall\eql@widthdata@calc@call
2230   \let\eql@widthdata@sepcall\eql@widthdata@calc@callsep
2231   \eql@widthdata@cells
2232 }
```

`\eql@widthdata@calc@callsep` Callback for each intercolumn space.

```
2233 \def\eql@widthdata@calc@callsep{%
2234   \advance\eql@line@possep@\@ne
2235 }
```

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

```

2236 \def\eq@widthdata@calc@call{%
2237   \advance\eq@column@\@ne
2238   \ifnum\eq@totalcolumns@=\@ne
2239     \dimen@\eq@totalwidth@
2240   \else
2241     \dimen@\eq@colwidth@get\eq@column@\relax
2242   \fi
2243   \ifdim\eq@cellwidth@>\z@
2244     \ifdim\eq@line@width@=\z@
2245       \eq@line@avail@\eq@line@pos@
2246       \eq@line@availsep@\eq@line@possep@
2247       \ifcase\eq@shape@pos@
2248       \or
2249         \advance\eq@line@avail@\dimexpr
2250           (\dimen@-\eq@cellwidth@+\eq@centeroffset@)/\tw@\relax
2251       \or
2252         \advance\eq@line@avail@\dimexpr\dimen@-\eq@cellwidth@\relax
2253       \fi
2254       \advance\eq@line@avail@\eq@shape@amount@
2255     \fi
2256     \eq@line@width@\eq@line@pos@
2257     \eq@line@widthsep@\eq@line@possep@
2258     \ifcase\eq@shape@pos@
2259     \advance\eq@line@width@\eq@cellwidth@
2260   \or
2261     \advance\eq@line@width@\dimexpr
2262       (\dimen@+\eq@cellwidth@+\eq@centeroffset@)/\tw@\relax
2263   \or
2264     \advance\eq@line@width@\dimen@
2265   \fi
2266   \advance\eq@line@width@\eq@shape@amount@
2267 \fi
2268 \advance\eq@line@pos@\dimen@
2269 }

```

## I.4 Best Line Selection

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

```

2270 \let\eq@numbering@best@auto\eq@false

```

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

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

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

```

2271 \newcount\eq@numbering@best@row@

```

```

2272 \newdimen\eq@numbering@best@space@

```

```

2273 \let\eq@numbering@best@use\eq@false

```

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

```

2274 \def\eq@numbering@best@find{%
2275   \eq@numbering@best@row@\z@
2276   \eq@numbering@best@space@\z@
2277   \eq@widthdata@for{%
2278     \eq@widthdata@calc
2279     \ifdefined\eq@tagsleft
2280       \dimen@\eq@line@avail@
2281     \else
2282       \dimen@\dimexpr\eq@totalwidth@-\eq@line@width@\relax

```

```

2283 \fi
2284 \ifdim\dimen@>\eql@numbering@best@space@
2285 \eql@numbering@best@row@\eql@row@
2286 \eql@numbering@best@space@\dimen@
2287 \fi
2288 }%
2289 \ifnum\eql@numbering@best@row@>\z@
2290 \eql@numbering@target@\eql@numbering@best@row@
2291 \fi
2292 }

```

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

```

2293 \def\eql@numbering@best@test{%
2294 \eql@widthdata@get\eql@numbering@target@
2295 \eql@widthdata@calc
2296 \ifdefined\eql@tagsleft
2297 \dimen@\dimexpr\eql@line@avail@
2298 +\eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax
2299 \else
2300 \dimen@\dimexpr\displaywidth-\eql@line@width@
2301 -\eql@marginleft@-\eql@line@widthsep@\eql@colsep@\relax
2302 \fi
2303 \ifdim\dimen@<\eql@tagwidth@single@
2304 \let\eql@numbering@best@use\eql@true
2305 \fi
2306 }

```

@numbering@best@eval **TODO:** describe

```

2307 \def\eql@numbering@best@eval{%
2308 \ifdefined\eql@numbering@best@auto
2309 \ifdefined\eql@numbering@best@use\else
2310 \ifnum\eql@numbering@target@>\z@
2311 \eql@numbering@best@test
2312 \fi
2313 \fi
2314 \fi
2315 \ifdefined\eql@numbering@best@use
2316 \eql@numbering@best@find
2317 \fi
2318 }

```

## I.5 Tag Margin

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

djust@calc@tagmargin

```

2319 \def\eql@adjust@calc@tagmargin{%
2320 \ifdefined\eql@tagmargin@val
2321 \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2322 \else
2323 \eql@tagmargin@\eql@tagwidth@max@
2324 \ifdim\eql@tagmargin@>\z@
2325 \advance\eql@tagmargin@-\eql@tagsepmin@
2326 \fi

```

```

2327 \fi

2328 \dimen@ \eq@tagrows@ \p@
2329 \ifnum \eq@totalrows@ = \@ne
2330   \ifnum \eq@tagrows@ = \@ne
2331     \advance \dimen@ 1sp \relax
2332   \fi
2333 \fi
2334 \ifdim \dimen@ > \eq@totalrows@ \eq@tagmargin@ratio@ \else
2335   \eq@tagmargin@ \z@
2336 \fi

2337 \@tempdima \dimexpr \displaywidth
2338   - \eq@totalwidth@ - \eq@intercolumns@ \eq@colsepmin@ \relax
2339 \@tempdimb \dimexpr \@tempdima - \tw@ \eq@tagmargin@ \relax
2340 \ifdim \@tempdimb > \z@
2341   \ifdim \eq@tagmargin@ > threshold \@tempdima < \@tempdimb
2342     \eq@tagmargin@ \z@
2343   \fi
2344 \fi
2345 }

```

## I.6 Single Column

eq@adjust@calc@lines

```

2346 \def \eq@adjust@calc@lines{%
2347   \eq@totalcolumns@ \@ne
2348   \eq@intercolumns@ \z@
2349   \eq@colsep@ \z@
2350   \ifdefined \eq@layoutleft
2351     \eq@marginleft@ \glueexpr \eq@layoutleftmargin@ \relax
2352     \eq@marginleft@min@ \glueexpr \eq@layoutleftmarginmin@ \relax
2353     \ifdim \eq@marginleft@ < \eq@marginleft@min@
2354       \eq@marginleft@ \eq@marginleft@min@
2355     \fi
2356     \dimen@ \glueexpr \eq@layoutleftmarginmax@ \relax
2357     \ifdim \eq@marginleft@ > \dimen@
2358       \eq@marginleft@ \dimen@
2359     \fi
2360     \eq@marginright@ \z@
2361     \eq@centeroffset@ \z@
2362   \else
2363     \eq@adjust@calc@tagmargin
2364     \ifdefined \eq@paddingleft@val
2365       \eq@marginleft@ \dimexpr
2366         (\displaywidth - \eq@totalwidth@ - \eq@tagmargin@) / \tw@
2367         - \glueexpr \eq@paddingleft@val \relax \relax
2368       \ifdim \eq@marginleft@ < \z@
2369         \eq@marginleft@ \z@
2370       \fi
2371     \else
2372       \eq@marginleft@ \z@
2373     \fi
2374     \ifdefined \eq@paddingright@val
2375       \eq@marginright@ \dimexpr
2376         (\displaywidth - \eq@totalwidth@ - \eq@tagmargin@) / \tw@
2377         - \glueexpr \eq@paddingright@val \relax \relax

```

```

2378     \ifdim\eql@marginright@<\z@
2379     \eql@marginright@\z@
2380     \fi
2381   \else
2382     \eql@marginright@\z@
2383     \fi
2384   \ifdim\eql@tagmargin@>\z@
2385     \ifdefined\eql@tagsleft
2386       \ifdim\eql@marginleft@<\eql@tagsepmin@
2387         \eql@marginleft@\eql@tagsepmin@
2388         \fi
2389       \advance\eql@marginleft@\eql@tagmargin@
2390       \advance\eql@centeroffset@\eql@tagmargin@
2391     \else
2392       \ifdim\eql@marginright@<\eql@tagsepmin@
2393         \eql@marginright@\eql@tagsepmin@
2394         \fi
2395       \advance\eql@marginright@\eql@tagmargin@
2396       \advance\eql@centeroffset@-\eql@tagmargin@
2397       \fi
2398     \fi
2399     \eql@marginleft@min@\z@
2400     \eql@centeroffset@\dimexpr\eql@marginright@-\eql@marginleft@
2401     \ifdefined\eql@tagsleft+ \else- \fi\eql@tagmargin@\relax
2402   \fi

2403   \eql@totalwidth@\dimexpr\displaywidth
2404   -\eql@marginleft@-\eql@marginright@\relax
2405 }

```

## I.7 Multiple Columns

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

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

```

2406 \def\eql@adjust@calc@columns{%

```

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

```

2407   \ifnum\eql@totalcolumns@<\thr@@
2408     \eql@totalcolumns@\tw@
2409     \let\eql@columns@fulllength\eql@false
2410   \fi

```

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

```

2411   \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax

```

Evaluate the minimum intercolumn space which we will need often:

```

2412   \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax

```

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

```

2413 \ifdefined\eql@layoutleft
2414   \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2415   \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2416   \ifdim\eql@marginleft@<\eql@marginleft@min@
2417     \eql@marginleft@\eql@marginleft@min@
2418   \fi
2419 \else

```

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

```

2420   \eql@adjust@calc@tagmargin
2421   \ifdefined\eql@columns@fulllength
2422     \ifdim\eql@tagmargin@>\z@
2423       \advance\eql@tagmargin@\eql@tagsepmin@
2424     \fi
2425   \fi
2426   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2427     -\eql@intercolumns@\eql@colsepmin@\relax
2428     \eql@tagmargin@\z@
2429   \fi
2430   \eql@marginleft@min@\z@
2431 \fi

```

Compute the intercolumn space \eql@colsep@:

```

2432 \ifnum\eql@intercolumns@>\z@

```

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

```

2433   \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2434   \ifdefined\eql@layoutleft
2435     \advance\eql@colsep@-\eql@marginleft@
2436   \else
2437     \advance\eql@colsep@-\eql@tagmargin@
2438   \fi
2439   \count@\eql@intercolumns@
2440   \ifdefined\eql@columns@fulllength\else
2441     \ifdefined\eql@layoutleft
2442       \advance\count@\@ne
2443     \else
2444       \advance\count@\tw@
2445     \fi
2446   \fi
2447   \divide\eql@colsep@\count@

```

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

```

2448   \ifdim\eql@colsep@<\eql@colsepmin@
2449     \eql@colsep@\eql@colsepmin@
2450   \else
2451     \ifdefined\eql@columns@fulllength\else
2452       \dimen@\glueexpr\eql@colsepmax@val\relax
2453       \ifdim\eql@colsep@>\dimen@
2454         \eql@colsep@\dimen@
2455       \fi
2456     \fi
2457   \fi

```



2458 \else

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

2459 \eq@colsep@eq@colsepmin@  
2460 \fi

Compute the left margin \eq@marginleft@ depending on the layout:

2461 \ifdefined\eq@layoutleft

Set the default value:

2462 \ifdim\eq@colsep@=\eq@colsepmin@

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

2463 \dimen@ \dimexpr \displaywidth - \eq@totalwidth@  
2464 - \eq@intercolumns@ \eq@colsep@ \relax  
2465 \ifdim \dimen@ < \eq@marginleft@  
2466 \ifdim \dimen@ < \eq@marginleft@min@  
2467 \eq@marginleft@ \eq@marginleft@min@  
2468 \else  
2469 \eq@marginleft@ \dimen@  
2470 \fi  
2471 \fi  
2472 \fi  
2473 \else

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

2474 \ifdefined\eq@columns@fulllength  
2475 \eq@marginleft@ \z@

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

2476 \else  
2477 \eq@marginleft@ \dimexpr (\displaywidth - \eq@totalwidth@  
2478 - \eq@intercolumns@ \eq@colsep@ - \eq@tagmargin@) / \tw@ \relax  
2479 \ifdim \eq@marginleft@ < \z@  
2480 \eq@marginleft@ \z@  
2481 \fi  
2482 \fi

Add tag margin in case of left tags:

2483 \ifdefined\eq@tagsleft  
2484 \advance \eq@marginleft@ \eq@tagmargin@  
2485 \fi  
2486 \fi

Find the best row for tag placement:

2487 \eq@numbering@best@eval

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

2488 \ifdefined\eq@tagsleft  
2489 \let \eq@adjust@columns@test \eq@adjust@columns@test@tagsleft

```

2490 \else
2491   \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2492 \fi

```

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

```

2493 \ifnum\eql@numbering@target@<\z@
2494   \eql@widthdata@for{%
2495     \ifdim\eql@tagwidth@>\z@
2496       \eql@widthdata@calc
2497       \eql@adjust@columns@test
2498     \fi
2499   }%
2500 \else
2501   \ifnum\eql@numbering@target@>\z@
2502     \ifnum\eql@numbering@target@>\eql@totalrows@ \else
2503       \eql@widthdata@get\eql@numbering@target@
2504       \eql@tagwidth@\eql@tagwidth@single@
2505       \eql@widthdata@calc
2506       \eql@adjust@columns@test
2507     \fi
2508   \fi
2509 \fi

```

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

```

2510 \advance\eql@totalwidth@\dimexpr
2511   \eql@intercolumns@\eql@colsep@+\eql@marginleft@\relax
2512 }

```

### Placement for Right Tags.

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

```

2513 \def\eql@adjust@columns@test@tagsright{%

```

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

```

2514   \@tempdima\dimexpr\displaywidth-\eql@line@width@-\eql@tagwidth@\relax

```

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

```

2515   \ifdim\@tempdima<\dimexpr
2516     \eql@marginleft@+\eql@line@widthsep@\eql@colsep@\relax

```

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

```

2517   \ifdim\@tempdima<\dimexpr
2518     \eql@marginleft@min@+\eql@line@widthsep@\eql@colsepmin@\relax\else

```

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

```

2519     \eql@adjust@columns@modify@tagsright
2520   \fi
2521 \fi
2522 }

```

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

```
2523 \def\eql@adjust@columns@modify@tagsright{%
```

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

```
2524 \ifnum\eql@line@widthsep@>\z@
2525   \dimen@ \@tempdima
2526   \count@ \eql@line@widthsep@
2527   \ifdefined\eql@layoutleft
2528     \advance\dimen@-\eql@marginleft@
2529   \else
2530     \ifdefined\eql@columns@fulllength\else
2531       \advance\count@\@ne
2532     \fi
2533   \fi
2534   \divide\dimen@\count@
```

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

```
2535   \ifdim\dimen@<\eql@colsep@
2536     \ifdim\dimen@<\eql@colsepmin@
2537       \eql@colsep@\eql@colsepmin@
2538     \else
2539       \eql@colsep@\dimen@
2540     \fi
2541   \fi
2542 \fi
```

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

```
2543 \dimen@\dimexpr\@tempdima-\eql@line@widthsep@\eql@colsep@\relax
2544 \ifdim\eql@marginleft@>\dimen@
2545   \eql@marginleft@\dimen@
2546 \fi
2547 }
```

## Placement for Left Tags.

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

```
2548 \def\eql@adjust@columns@test@tagsleft{%
```

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

```
2549 \count@\numexpr\eql@intercolumns@-\eql@line@availsep@\relax
2550 \@tempdima\dimexpr\eql@tagwidth@-\eql@line@avail@\relax
```

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

```
2551 \ifdim\@tempdima>\dimexpr
2552   \eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax
```

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

```

2553 \ifdim\eq@tagwidth@<
2554 \ifdefined\eq@layoutleft
2555 \glueexpr\eq@layoutleftmarginmax\relax
2556 \else
2557 \displaywidth
2558 \fi

```

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

```

2559 \ifdim\@tempdima>\dimexpr
2560 \displaywidth-\eq@totalwidth@-\count@\eq@colsepmin@\relax\else

```

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

```

2561 \eq@adjust@columns@modify@tagsleft
2562 \fi
2563 \fi
2564 \fi
2565 }

```

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

```

2566 \def\eq@adjust@columns@modify@tagsleft{%

```

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

```

2567 \ifnum\count@>\z@
2568 \dimen@ \dimexpr \displaywidth-\eq@totalwidth@-\@tempdima \relax
2569 \ifdefined\eq@columns@fulllength\else
2570 \advance\count@ \@ne
2571 \fi
2572 \divide \dimen@ \count@

```

If smaller, reduce the intercolumn separation, but make sure to not exceed the minimum allowed value. Also adjust the left margin to keep the right margin fixed.

```

2573 \ifdim\dimen@<\eq@colsep@
2574 \ifdim\dimen@<\eq@colsepmin@
2575 \dimen@ \eq@colsepmin@
2576 \fi
2577 \advance\dimen@-\eq@colsep@
2578 \advance\eq@marginleft@-\eq@intercolumns@\dimen@
2579 \advance\eq@colsep@\dimen@
2580 \fi
2581 \fi

```

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

```

2582 \dimen@ \dimexpr \@tempdima-\eq@line@availsep@\eq@colsep@ \relax
2583 \ifdim\eq@marginleft@<\dimen@
2584 \eq@marginleft@ \dimen@
2585 \fi
2586 }

```

## J Single Column Arrangement

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

### J.1 Supporting Definitions

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

```
2587 \ifdefined\inf@bad\else%
2588   \newcount\inf@bad
2589   \inf@bad1000000\relax
2590 \fi
```

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

```
2591 \let\eq@restore@hfuzz\@empty
2592 \def\eq@save@hfuzz{\edef\eq@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

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

```
2593 \newcount\eq@alignbadness@
2594 \newcount\eq@tagbadness@
2595 \newcount\eq@arrange@badness@
2596 \eq@alignbadness@\inf@bad
2597 \eq@tagbadness@\inf@bad
```

### J.2 Arrangement Methods

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

```
2598 \def\eq@arrange@try#1{%
2599   \ifdim#1>\dimexpr\displaywidth-\eq@cellwidth@\relax
2600     \setbox\eq@cellbox@\hbox to\displaywidth{%
2601       \unhbox\eq@cellbox@\unkern\kern#1}%
2602     \eq@arrange@badness@\badness
2603   \else
2604     \eq@arrange@badness@\m@ne
2605   \fi
2606 }
```

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

```
2607 \def\eq@arrange@print#1#2{%
2608   \ifnum\eq@raisetag@mode@>\z@
2609     \ifdefined\eq@tagsleft
2610       \eq@display@firstavail@set\z@
2611       \eq@tagbox@print@tagsleft
2612     \fi
```

```

2613 \fi
2614 \eql@restore@hfuzz
2615 \hbox to\displaywidth{%
2616   #1%
2617   \unhbox\eql@cellbox@\unkern
2618   #2%
2619   \eql@tagging@mathaddlast
2620 }%
2621 \ifnum\eql@raisetag@mode@>\z@
2622   \ifdefined\eql@tagsleft\else
2623     \eql@tagbox@print@tagsright
2624   \fi
2625 \fi
2626 }

```

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

```

2627 \def\eql@arrange@print@alignleft#1#2{%
2628   \eql@display@firstavail@set{#1}%
2629   \eql@arrange@print{\kern#1}{\kern#2}%
2630 }

2631 \def\eql@arrange@print@alignright#1#2{%
2632   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
2633   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
2634 }

2635 \def\eql@arrange@print@aligncenter#1{%
2636   \eql@display@firstavail@set{\dimexpr
2637     (\displaywidth-\eql@cellwidth@+1)/2\relax}%
2638   \ifdim#1>\z@
2639     \eql@arrange@print{\kern#1\hfil}{}%
2640   \else
2641     \eql@arrange@print{\hfil}{\kern-#1}%
2642   \fi
2643 }

```

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

```

2644 \def\eql@arrange@init{%
2645   \eql@save@hfuzz
2646   \hfuzz\maxdimen
2647   \eql@shape@select
2648 }

```

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

```

2649 \def\eql@arrange@print@line{%
2650   \eql@tagging@tagaddbox
2651   \csname eql@arrange%
2652     @ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
2653     @init\endcsname
2654   \csname eql@arrange%
2655     @ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi

```

```

2656     \@ifnum\eql@raisetag@mode@=\tw@
2657     \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
2658     notag\fi\endcsname
2659 }

```

### J.3 Central Alignment

**TODO:** describe

```

2660 \def\eql@arrange@aligncenter@init{%
2661   \eql@tagging@aligncenter
2662   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
2663   +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
2664 }

```

**TODO:** describe

```

2665 \def\eql@arrange@aligncenter@notag{%
2666   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>
2667     \ifdim\eql@line@offset@<\eql@marginleft@min@
2668       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
2669     \else
2670       \eql@line@offset@
2671     \fi
2672   \eql@arrange@print@aligncenter\eql@line@offset@
2673 \else
2674   \ifdim\eql@line@offset@<\eql@marginleft@min@
2675     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
2676   \else
2677     \eql@arrange@print@alignright\eql@marginleft@min@\z@
2678   \fi
2679 \fi
2680 }

```

**TODO:** describe

```

2681 \def\eql@arrange@aligncenter@tagsright{%
2682   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>
2683     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
2684       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
2685     \else
2686       \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
2687     \fi
2688   \eql@arrange@print@aligncenter\eql@line@offset@
2689 \else
2690   \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
2691   \ifnum\eql@arrange@badness@<\eql@tagbadness@
2692     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
2693       \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
2694     \else
2695       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
2696     \fi
2697   \else
2698     \eql@raisetag@mode@\@ne
2699     \eql@arrange@aligncenter@notag
2700   \fi
2701 \fi
2702 }

```

```

2703 \def\eqL@arrange@aligncenter@tagsleft{%
2704   \ifdim\eqL@tagwidth@>\eqL@marginleft@min@
2705     \ifdim\dimexpr\displaywidth-\eqL@cellwidth@>\relax
2706       \ifdim\eqL@line@offset@<\eqL@tagwidth@
2707         \dimexpr\tw@\eqL@tagwidth@-\eqL@line@offset@>\relax
2708       \else
2709         \eqL@line@offset@
2710       \fi
2711     \eqL@arrange@print@aligncenter\eqL@line@offset@
2712   \else
2713     \eqL@arrange@try\eqL@tagwidth@
2714     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
2715       \ifdim\eqL@line@offset@<\eqL@tagwidth@
2716         \eqL@arrange@print@alignleft\eqL@tagwidth@\z@
2717       \else
2718         \eqL@arrange@print@alignright\eqL@tagwidth@\z@
2719       \fi
2720     \else
2721       \eqL@raisetag@mode@\@ne
2722       \eqL@arrange@aligncenter@notag
2723     \fi
2724   \fi
2725 \else
2726   \eqL@arrange@aligncenter@notag
2727 \fi
2728 }

```

## J.4 Left Alignment

```

2729 \def\eqL@arrange@alignleft@init{%
2730   \eqL@tagging@alignleft
2731   \eqL@line@offset@\dimexpr\eqL@marginleft@+\eqL@shape@amount@>\relax
2732   \ifdim\eqL@line@offset@<\eqL@marginleft@min@
2733     \eqL@line@offset@\eqL@marginleft@min@
2734   \fi
2735 }

2736 \def\eqL@arrange@alignleft@notag{%
2737   \ifdim\eqL@line@offset@>\eqL@marginleft@min@
2738     \eqL@arrange@try\eqL@line@offset@
2739     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
2740       \eqL@arrange@print@alignleft\eqL@line@offset@\z@
2741     \else
2742       \eqL@arrange@print@alignright\eqL@marginleft@min@\z@
2743     \fi
2744   \else
2745     \eqL@arrange@print@alignleft\eqL@marginleft@min@\z@
2746   \fi
2747 }

2748 \def\eqL@arrange@alignleft@tagsright{%
2749   \eqL@arrange@try{\dimexpr\eqL@line@offset@+\eqL@tagwidth@>\relax}%
2750   \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
2751     \eqL@arrange@print@alignleft\eqL@line@offset@\eqL@tagwidth@
2752   \else
2753     \ifdim\eqL@line@offset@>\eqL@marginleft@min@
2754       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@>\relax}%
2755     \fi
2756     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@

```



```

2757 \eq@arrange@print@alignright\eq@marginleft@min@\eq@tagwidth@
2758 \else
2759 \eq@raisetag@mode@\@ne
2760 \eq@arrange@alignleft@notag
2761 \fi
2762 \fi
2763 }

2764 \def\eq@arrange@alignleft@tagsleft{%
2765 \ifdim\eq@tagwidth@>\eq@marginleft@min@
2766 \ifdim\eq@line@offset@>\eq@tagwidth@
2767 \eq@arrange@try\eq@line@offset@
2768 \ifnum\eq@arrange@badness@<\eq@alignbadness@
2769 \eq@arrange@print@alignleft\eq@line@offset@z@
2770 \else
2771 \eq@arrange@try\eq@tagwidth@
2772 \ifnum\eq@arrange@badness@<\eq@tagbadness@
2773 \eq@arrange@print@alignright\eq@tagwidth@z@
2774 \else
2775 \eq@raisetag@mode@\@ne
2776 \eq@arrange@print@alignright\eq@marginleft@min@z@
2777 \fi
2778 \fi
2779 \else
2780 \eq@arrange@try\eq@tagwidth@
2781 \ifnum\eq@arrange@badness@<\eq@tagbadness@
2782 \eq@arrange@print@alignleft\eq@tagwidth@z@
2783 \else
2784 \eq@raisetag@mode@\@ne
2785 \eq@arrange@alignleft@notag
2786 \fi
2787 \fi
2788 \else
2789 \eq@arrange@alignleft@notag
2790 \fi
2791 }

```

## J.5 Right Alignment

```

2792 \def\eq@arrange@alignright@init{%
2793 \eq@tagging@alignright
2794 \eq@line@offset@\dimexpr\eq@marginright@-\eq@shape@amount@\relax
2795 \ifdim\eq@line@offset@<z@
2796 \eq@line@offset@z@
2797 \fi
2798 }

```

**TODO:** describe

```

2799 \def\eq@arrange@alignright@notag{%
2800 \ifdim\eq@line@offset@>z@
2801 \eq@arrange@try{\dimexpr\eq@marginleft@min@+\eq@line@offset@\relax}%
2802 \ifnum\eq@arrange@badness@<\eq@alignbadness@
2803 \eq@arrange@print@alignright\eq@marginleft@min@\eq@line@offset@
2804 \else
2805 \eq@arrange@print@alignleft\eq@marginleft@min@z@
2806 \fi
2807 \else
2808 \eq@arrange@print@alignright\eq@marginleft@min@z@
2809 \fi

```

2810 }

**TODO:** describe

```
2811 \def\eql@arrange@alignright@tagsright{%
2812   \ifdim\eql@line@offset@>\eql@tagwidth@
2813     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
2814     \ifnum\eql@arrange@badness@<\eql@alignbadness@
2815       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
2816     \else
2817       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
2818       \ifnum\eql@arrange@badness@<\eql@tagbadness@
2819         \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
2820       \else
2821         \eql@raisetag@mode@\@ne
2822         \eql@arrange@print@alignleft\eql@marginleft@min@\z@
2823       \fi
2824     \fi
2825   \else
2826     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
2827     \ifnum\eql@arrange@badness@<\eql@tagbadness@
2828       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
2829     \else
2830       \eql@raisetag@mode@\@ne
2831       \eql@arrange@alignright@notag
2832     \fi
2833   \fi
2834 }
```

**TODO:** describe

```
2835 \def\eql@arrange@alignright@tagsleft{%
2836   \ifdim\eql@tagwidth@>\eql@marginleft@min@
2837     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
2838     \ifnum\eql@arrange@badness@<\eql@alignbadness@
2839       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
2840     \else
2841       \ifdim\eql@line@offset@>\z@
2842         \eql@arrange@try\eql@tagwidth@
2843       \fi
2844       \ifnum\eql@arrange@badness@<\eql@tagbadness@
2845         \eql@arrange@print@alignleft\eql@tagwidth@\z@
2846       \else
2847         \eql@raisetag@mode@\@ne
2848         \eql@arrange@alignright@notag
2849       \fi
2850     \fi
2851   \else
2852     \eql@arrange@alignright@notag
2853   \fi
2854 }
```

## K Equations Box Environment

**TODO:** outline sequence of calls

**TODO:** describe

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

**TODO:** vspace?!

## K.1 Line Breaks

`\eql@box@cr`

```

2855 \protected\def\eql@box@cr{%
2856   \eql@ampprotecttwo{\eql@ifnextchar@tight[]\eql@box@cr@skip\eql@box@cr@
2857 }
2858 \def\eql@box@cr@{%
2859   \eql@punct@apply@line
2860   \eql@hook@lineout
2861   \eql@box@lastcell
2862   \cr
2863 }
2864 \def\eql@box@cr@skip[#1]{%
2865   \eql@box@cr@
2866   \noalign{%
2867     \vskip\glueexpr#1\relax
2868   }%
2869 }
```

## K.2 Stacked Mode

```

2870 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}
```

**TODO:** templates

```

2871 \def\eql@box@open@stacked{%
2872   \eql@shape@align@enable
2873   \let\eql@box@lastcell\eql@box@lastcell@stacked
2874   \everycr{\noalign{%
2875 (dev)\eql@dev{starting line \the\eql@row@}%
2876   \global\advance\eql@row@\@ne
2877   }}%
2878   \tabskip\z@skip
2879   \halign\bgroup
2880     &%
2881     \global\let\eql@cell@container\@empty
2882     \setbox\eql@cellbox@\hbox{%
2883       \eql@strut@cell
2884       \@lign
2885       $\m@th\displaystyle
2886       \eql@hook@colin
2887       ##%
2888       \eql@punct@apply@col
2889       \eql@hook@colout
2890       \eql@tagging@mathsave
2891       $%
2892       \eql@tagging@mathaddlast
2893     }%
2894     \ifdefined\eql@shape@lastrow
2895       \eql@totalrows@\eql@row@
2896     \fi
2897     \eql@shape@eval
2898     \eql@cell@container
2899     \ifdefined\eql@frame@cmd
2900       \ifcase\eql@shape@pos@
2901         \eql@frame@measure
2902         \advance\eql@shape@amount@-\eql@frame@margin@
2903       \or\or
```

```

2904         \eql@frame@measure
2905         \advance\eql@shape@amount@+\eql@frame@margin@
2906     \fi
2907     \eql@frame@print
2908 \fi
2909 \ifcase\eql@shape@pos@
2910     \kern\eql@shape@amount@
2911     \box\eql@cellbox@
2912     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
2913     -\eql@shape@amount@+\@flushglue\relax
2914     \eql@tagging@alignleft
2915 \or
2916     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
2917     \box\eql@cellbox@
2918     \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
2919     \eql@tagging@aligncenter
2920 \or
2921     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
2922     +\eql@shape@amount@+\@flushglue\relax
2923     \box\eql@cellbox@
2924     \kern-\eql@shape@amount@
2925     \eql@tagging@alignright
2926 \fi
2927     \tabskip\eql@colsep@\relax
2928 \crrc
2929 \noalign{%
2930     \global\let\eql@shape@lastrow\eql@false
2931     \eql@hook@blockbefore
2932 }%
2933 \eql@hook@blockin
2934 }
2935 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

### K.3 Aligned Mode

```

2936 \def\eql@box@lastcell@odd{%
2937     &\omit
2938     \eql@prevwidth@\wd\eql@cellbox@
2939     \let\eql@frame@cmd\eql@frame@prevcmd
2940     \ifdefined\eql@frame@cmd
2941         \eql@frame@measure
2942         \advance\eql@prevwidth@\eql@frame@margin@
2943         \eql@frame@print
2944     \fi
2945     \kern-\eql@prevwidth@
2946     \unhbox\eql@cellbox@
2947     \hfil
2948     &\omit\kern-\eql@colsep@
2949 }%
2950 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}
2951 \def\eql@box@open@aligned{%
2952 % \TODO templates
2953     \eql@shape@align@disable
2954     \let\eql@box@lastcell\@empty
2955     \everycr{\noalign{%
2956 (dev)\eql@dev{starting new line}%
2957 }}%
2958     \tabskip\z@skip

```

```

2959 \halign\bgroup
2960   &%
2961   \let\eql@box@lastcell\eql@box@lastcell@odd
2962   \global\let\eql@cell@container\@empty
2963   \global\setbox\eql@cellbox@\hbox{%
2964     \eql@strut@cell
2965     \@lign
2966     $\m@th\displaystyle
2967     \eql@hook@colin
2968     ##%
2969     \eql@class@innerleft
2970     \eql@hook@innerleft
2971     \eql@tagging@mathsave
2972     $%
2973     \eql@tagging@mathaddlast
2974   }%
2975   \eql@cell@container
2976   \hfil
2977   \kern\wd\eql@cellbox@
2978   \ifdefined\eql@frame@cmd
2979     \eql@frame@measure
2980     \kern\eql@frame@margin@
2981   \fi
2982   \global\let\eql@frame@prevcmd\eql@frame@cmd
2983   \tabskip\z@skip
2984   &%
2985   \eql@prevwidth@\wd\eql@cellbox@
2986   \let\eql@box@lastcell\eql@box@lastcell@even
2987   \let\eql@frame@cmd\eql@frame@prevcmd
2988   \global\let\eql@cell@container\@empty
2989   \setbox\eql@cellbox@\hbox{%
2990     \unhbox\eql@cellbox@
2991     \eql@strut@cell
2992     \@lign
2993     $\m@th\displaystyle
2994     \eql@hook@innerright
2995     \eql@class@innerright@sel
2996     ##%
2997     \eql@punct@apply@col
2998     \eql@hook@colout
2999     \eql@tagging@mathsave
3000     $%
3001     \eql@tagging@mathaddlast
3002   }%
3003   \eql@cell@container
3004   \ifdefined\eql@frame@cmd
3005     \eql@frame@measure
3006     \advance\eql@prevwidth@\eql@frame@margin@
3007     \eql@frame@print
3008   \fi
3009   \kern-\eql@prevwidth@
3010   \unhbox\eql@cellbox@
3011   \hfil
3012   \tabskip\eql@colsep@\relax
3013 \crrr
3014 \noalign{%
3015   \eql@hook@blockbefore
3016 }%

```

```

3017 \eq@hook@blockin
3018 }
3019 \def\eq@mode@aligned{\let\eq@box@open\eq@box@open@aligned}

```

## K.4 Main

```

3020 \let\eq@box@box\vcenter
3021 \let\eq@box@open\@undefined
3022 \let\eq@box@frame\@firstofone
3023 \def\eq@box@wrap#1#2{\def\eq@box@frame##1{#1##1#2}}

```

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

```

3024 \def\eq@box@close{%
3025   \ifvmode\else
3026     \global\let\eq@shape@lastrow\eq@true
3027     \eq@punct@apply@block
3028     \eq@box@cr@
3029   \fi
3030   \noalign{%
3031     \eq@hook@blockafter
3032     \global\let\eq@shape@lastrow\eq@false
3033   }%
3034   \eq@tagging@tablesaveinner
3035 \egroup
3036 }

```

`\eq@box@vcenter`

```

3037 \def\eq@box@vcenter#1{%
3038   \ifmmode
3039     \vcenter{#1}%
3040   \else
3041     $\m@th\vcenter{#1}$%
3042   \fi
3043 }

```

`\eq@box@start`

```

3044 \let\eq@box@endmath\eq@false
3045 \def\eq@box@start{%
3046   \relax
3047   \ifmmode
3048     \let\eq@box@endmath\eq@false
3049   \else
3050     \let\eq@box@endmath\eq@true
3051     \expandafter$%$
3052   \fi
3053   \eq@box@processopt
3054   \eq@stack@save@box
3055   \let\eq@frame@cmd\@undefined
3056   \let\eq@layoutleft\eq@false
3057   \eq@row@z@
3058   \eq@totalrows@\@M
3059   \eq@shape@select
3060   \setbox\z@\ifx\eq@box@box\vcenter

```

```

3061 \expandafter\ vbox
3062 \else
3063 \expandafter\ eql@box@box
3064 \fi\bgroup
3065 \eql@display@nest
3066 \let\\eql@box@cr
3067 \eql@spread@set
3068 \eql@strut@make
3069 \eql@box@open
3070 }

```

`\eql@box@end`

```

3071 \def\eql@box@end{%
3072 \eql@box@close
3073 \egroup
3074 \eql@box@frame{%
3075 \ifdefined\eql@display@marginleft
3076 \hskip\glueexpr\eql@display@marginleft\relax
3077 \fi
3078 \ifx\eql@box@box\vcenter
3079 \eql@box@vcenter{\unvbox\z@}%
3080 \else
3081 \box\z@
3082 \fi
3083 \eql@tagging@tableaddinner
3084 \ifdefined\eql@display@marginright
3085 \hskip\glueexpr\eql@display@marginright\relax
3086 \fi
3087 }%
3088 \eql@stack@restore
3089 \ifdefined\eql@box@endmath
3090 \expandafter$%$
3091 \fi
3092 }

```

## K.5 Environment

`equationsbox` (*env.*)

```

3093 \newenvironment{equationsbox}{%
3094 (dev)\eql@dev@enterenv
3095 \eql@ampprotect\eql@box@testall\eql@box@start
3096 }{%
3097 \eql@box@end
3098 (dev)\eql@dev@leaveenv
3099 }

3100 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3101 \def\eql@box@parseopt{%
3102 % \ifx\eql@parseopt@token~%
3103 % \let\eql@parseopt@next\eql@parseopt@lines
3104 % \fi
3105 \ifx\eql@parseopt@token[%]
3106 \let\eql@parseopt@next\eql@parseopt@opt
3107 \fi
3108 \ifx\eql@parseopt@token=%
3109 \let\eql@parseopt@next\eql@parseopt@lines

```

```

3110 \fi
3111 \ifx\eql@parseopt@token%
3112   \let\eql@parseopt@next\eql@parseopt@columns
3113 \fi
3114 }

```

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

```

3115 \def\eql@box@processopt{%
3116   \let\eql@box@frame\@firstofone
3117   \let\eql@display@marginleft\@undefined
3118   \let\eql@display@marginright\@undefined
3119   \eql@nextopt@process{equationsbox}%
3120   \let\eql@punct@block\eql@punct@main
3121   \let\eql@punct@main\relax
3122   \eql@colsep@\glueexpr\eql@box@colsep\relax
3123   \ifdefined\eql@paddingleft@val
3124     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3125   \else
3126     \eql@paddingleft@\z@
3127   \fi
3128   \ifdefined\eql@paddingright@val
3129     \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3130   \else
3131     \eql@paddingright@\z@
3132   \fi
3133   \eql@indent@\glueexpr\eql@indent@val\relax
3134 }

```

## L Single-Line Equation

**TODO:** describe

### L.1 Native Mode

```

3135 \def\eql@single@start@native{%
3136   \eql@display@init
3137   \eql@display@print
3138   \let\raisetag\eql@raisetag@default
3139   \eql@shape@align@disable
3140   \eql@hook@eqin
3141   % \mathopen{}%
3142 }%

```

**TODO:** describe

```

3143 \def\eql@single@end@native{%
3144   % \mathclose{}%
3145   \if@eqnsw
3146     \ifdefined\eql@tagsleft
3147       \leqno
3148     \else
3149       \eqno
3150     \fi
3151     \eql@tags@container
3152     \eql@composetag@print
3153   \fi

```



```

3154 \eql@interline@container
3155 \eql@display@penalty
3156 \eql@display@vspace@native
3157 }%

```

## L.2 Print

```

3158 \def\eql@single@start@print{%
3159   \eql@display@init
3160   \eql@display@print
3161   \eql@shape@align@enable

3162   \eql@totalrows@ \@ne
3163   \eql@row@ \@ne
3164   \eql@arrange@init
3165   \global\let\eql@cell@container \@empty

3166   \prevgraf\numexpr\prevgraf+\@ne\relax
3167   \setbox\eql@cellbox@ \hbox\bgroup
3168     \eql@restore@hfuzz
3169     \eql@strut@cell
3170     $\m@th\displaystyle%$
3171     \eql@hook@eqin
3172 }

3173 \def\eql@single@end@print{%
3174   \eql@tagging@mathsave
3175   $%$
3176   \hfil
3177   \kern\z@
3178   \egroup
3179   \prevgraf\numexpr\prevgraf-\@ne\relax

3180   \eql@shape@eval
3181   \eql@cell@container

3182   \ifdefined\eql@frame@cmd
3183     \eql@frame@adjust
3184   \fi

3185   \eql@cellwidth@ \wd\eql@cellbox@
3186   \eql@line@height@ \ht\eql@cellbox@
3187   \eql@line@depth@ \dp\eql@cellbox@
3188   \eql@totalwidth@ \eql@cellwidth@

3189   \eql@tags@container
3190   \if@eqnsw
3191     \eql@tagbox@make\eql@composetag@print
3192     \eql@tagrows@ \@ne
3193     \ifnum\eql@raisetag@mode@=\@ne
3194       \eql@tagwidth@ \z@
3195     \fi
3196   \else
3197     \eql@numbering@warnunused
3198     \eql@tagwidth@ \z@
3199     \eql@tagrows@ \z@
3200   \fi
3201   \eql@tagwidth@max@ \eql@tagwidth@

3202   \eql@intercolumns@ \z@
3203   \eql@adjust@calc@lines

3204   \eql@display@halign@init{ }%
3205   \halign{##\crr

```

```

3206 \noalign{\eqldisplay@halign@start}%
3207 \if@eqnsw\else
3208 \eqldraisetag@mode@z@
3209 \fi
3210 \eqldarrange@print@line
3211 \cr
3212 \noalign{\eqldisplay@halign@end}%
3213 \eqldtagging@tablesavelines
3214 }%
3215 \eqldisplay@close
3216 }

```

## M Multi-Line with Single Column

**TODO:** outline sequence of calls

### M.1 Measure

**TODO:** describe

```

3217 \def\eqldlines@measure@line@begin{%
3218 (dev)\eqlddev{starting line \the\eqldrow}%
3219 \eqldnumbering@measure@line@begin
3220 \eqldhook@linein
3221 }

```

**TODO:** describe

```

3222 \def\eqldlines@measure@line@end{%
3223 \eqldpunct@apply@line
3224 \eqldhook@lineout
3225 }

```

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

```

3226 \def\eqldlines@measure@cell{%
3227 \ifdefined\eqldframe@cmd
3228 \ifcase\eqldshape@pos@
3229 \eqldframe@measure
3230 \advance\eqldshape@amount@-\eqldframe@margin@
3231 \or\or
3232 \eqldframe@measure
3233 \advance\eqldshape@amount@+\eqldframe@margin@
3234 \fi
3235 \eqldframe@print
3236 \fi
3237 \eqldcellwidth@wd\eqldcellbox@
3238 \eqldwidthdata@startrow
3239 \eqldwidthdata@savercell
3240 \kern\eqldcellwidth@
3241 }

```

\eqldlines@measure

```

3242 \def\eqldlines@measure{%
3243 (dev)\eqlddev@enter\eqldlines@measure
3244 \eqldmeasure@init\eqldlines@measure@line@begin\eqldlines@measure@line@end

```

```

3245 \eql@totalrows@\@M
3246 \eql@shape@select

3247 \setbox\z@\vbox{\halign{%
3248   \global\let\eql@cell@container\@empty
3249   \setbox\eql@cellbox@\hbox{%
3250     \@lign
3251     $\m@th\displaystyle
3252     \eql@hook@colin
3253     ##%
3254     \eql@punct@apply@col
3255     \eql@hook@colout
3256     $%
3257   }%
3258   \ifdefined\eql@shape@lastrow
3259     \eql@totalrows@\eql@row@
3260   \fi
3261   \eql@shape@eval
3262   \eql@cell@container
3263   \eql@lines@measure@cell
3264   \eql@measure@tag
3265 \crrr

3266 \noalign{%
3267   \global\let\eql@shape@lastrow\eql@false
3268   \eql@hook@blockbefore
3269 }%
3270 \eql@hook@blockin
3271 \eql@scan@body
3272 \ifvmode\else
3273   \global\let\eql@shape@lastrow\eql@true
3274   \eql@punct@apply@block
3275   \eql@hook@blockout
3276   \eql@display@endline
3277   \cr
3278 \fi
3279 \omit
3280 \cr
3281 \noalign{%
3282   \eql@hook@blockafter
3283   \global\let\eql@shape@lastrow\eql@false
3284 }%
3285 }}%

3286 \eql@measure@close

3287 \setbox\z@\vbox{%
3288   \unvbox\z@
3289   \unpenalty
3290   \global\setbox\@ne\lastbox
3291 }%
3292 \eql@totalwidth@\wd\@ne

3293 (dev)\eql@dev@leave\eql@lines@measure
3294 }

```

## M.2 Column Placement

**TODO:** describe Find the best row for tag placement:

```
3295 \def\eql@lines@adjust{%
3296   \eql@adjust@calc@lines
3297   \eql@numbering@best@eval
3298 }
```

## M.3 Print

**TODO:** describe

nes@print@line@begin

```
3299 \def\eql@lines@print@line@begin{%
3300 (dev)\eql@dev{starting line \the\eql@row}%
3301   \eql@numbering@print@line@begin
3302   \eql@hook@linein
3303 }
```

**TODO:** describe

```
3304 \def\eql@lines@print@line@end{%
3305   \eql@punct@apply@line
3306   \eql@hook@lineout
3307 }
```

**TODO:** describe

```
3308 \def\eql@lines@print@line@adjust{%
3309   \ifdefined\eql@frame@cmd
3310     \ifcase\eql@shape@pos@
3311       \eql@frame@measure
3312       \advance\eql@shape@amount@-\eql@frame@margin@
3313     \or\or
3314       \eql@frame@measure
3315       \advance\eql@shape@amount@+\eql@frame@margin@
3316     \fi
3317     \eql@frame@adjust
3318   \fi
3319   \eql@cellwidth@\wd\eql@cellbox@
3320   \eql@line@height@\ht\eql@cellbox@
3321   \eql@line@depth@\dp\eql@cellbox@
3322   \eql@numbering@print@line@eval
3323   \if@eqnsw
3324     \eql@tags@container
3325     \eql@tagbox@make\eql@composetag@print
3326   \else
3327     \eql@raisetag@mode@\z@
3328   \fi
3329   \eql@arrange@print@line
3330 }
```

**TODO:** describe

```
3331 \def\eql@lines@print{%
3332 (dev)\eql@dev@center\eql@lines@print
3333   \eql@arrange@init
3334   \eql@display@halign@init\eql@lines@print@line@begin
```

```

3335 \eqldisplay@halign@letcr\eqllines@print@line@end
3336 \tabskip\z@skip

3337 \halign{%
3338   \global\let\eql@cell@container\@empty
3339   \setbox\eql@cellbox@\hbox{%
3340     \eql@restore@hfuzz
3341     \eql@strut@cell
3342     \@lign
3343     $\m@th\displaystyle
3344     \eql@hook@colin
3345     ##%
3346     \eql@punct@apply@col
3347     \eql@hook@colout
3348     \eql@tagging@mathsave
3349     $%
3350     \hfil
3351     \kern\z@
3352   }%
3353   \eql@shape@eval
3354   \eql@cell@container
3355   \eqllines@print@line@adjust
3356 \cr

3357 \noalign{%
3358   \eqldisplay@halign@start
3359   \eql@numbering@print@block@begin
3360   \eql@hook@blockbefore
3361 }%
3362 \eql@hook@blockin
3363 \eql@scan@body
3364 \ifvmode\else
3365   \relax
3366   \eql@punct@apply@block
3367   \eql@hook@blockout
3368   \eqldisplay@endline
3369   \cr
3370 \fi
3371 \noalign{%
3372   \eql@hook@blockafter
3373   \eqldisplay@halign@end
3374 (dev)\eql@dev@leave\eqllines@print
3375 }%
3376 \eql@tagging@tablesavelines
3377 }%
3378 }

```

## N Multi-Line with Multiple Columns

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

### N.1 Support

**TODO:** describe

\eql@columns@add@amp  
@columns@completerow

```

3379 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3380 \def\eql@columns@completerow{%
3381   \count@\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
3382   \edef\eql@tmp{%
3383     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3384   \eql@tmp
3385 }

3386 \def\eql@columns@overfull{%
3387   \dimen@\eql@line@width@
3388   \advance\dimen@-\hfuzz
3389   \ifdim\dimen@>\displaywidth
3390     \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3391     \wd\z@\z@
3392     \ht\z@\eql@line@height@
3393     \dp\z@\eql@line@depth@
3394     \box\z@
3395   \fi
3396 }

```

## N.2 Transpose

**TODO:** describe

**TODO:** describe

```

3397 \let\eql@transpose@active\eql@false
3398 \def\eql@transpose@end{\eql@transpose@end}
3399 \def\eql@transpose@skip{&\eqnpunct{}}
3400 \def\eql@transpose@complete{%
3401   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

**TODO:** describe

```

3402 \def\eql@transpose{%
3403   \eql@totalcolumns@\z@
3404   \eql@totalrows@\z@
3405   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
3406   \eql@scan@reg@{}%
3407   \eql@row@\z@
3408   \eql@transpose@output@row
3409 }

```

**TODO:** describe

```

3410 \def\eql@transpose@save@col#1{%
3411   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
3412     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

**TODO:** describe

```

3413 \def\eql@transpose@scan@col#1\&{%
3414   \def\@tempa{#1}%
3415   \ifx\@tempa\eql@transpose@end\else
3416     \advance\eql@totalcolumns@\@ne
3417     \eql@row@\z@
3418     \let\eql@transpose@data@col@\empty
3419     \eql@transpose@scan@row#1\\eql@transpose@end\\%
3420     \ifnum\eql@row@>\eql@totalrows@

```

```

3421     \eql@totalrows@eql@row@
3422     \fi
3423     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
3424     \expandafter\eql@transpose@scan@col
3425     \fi
3426 }

```

**TODO:** describe

```

3427 \def\eql@transpose@append@row#1{%
3428     \advance\eql@row@\@ne
3429     \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}

```

**TODO:** describe

```

3430 \def\eql@transpose@scan@row#1\\{%
3431     \def\@tempa{#1}%
3432     \ifx\@tempa\eql@transpose@end\else
3433         \ifx\eql@transpose@active+
3434             \eql@transpose@scan@cell#1\eql@transpose@end&%
3435         \else
3436             \eql@transpose@append@row{#1}%
3437         \fi
3438     \expandafter\eql@transpose@scan@row
3439     \fi
3440 }

```

**TODO:** describe

```

3441 \def\eql@transpose@scan@cell#1&#2{%
3442     \def\@tempa{#2}%
3443     \ifx\@tempa\eql@transpose@end
3444         \eql@transpose@append@row{#1}%
3445     \else
3446         \eql@transpose@append@row{#1&#2}%
3447     \expandafter\eql@transpose@scan@cell@next
3448     \fi
3449 }

```

**TODO:** describe

```

3450 \def\eql@transpose@scan@cell@next#1&{%
3451     \def\@tempa{#1}%
3452     \ifx\@tempa\eql@transpose@end\else
3453         \eql@transpose@append@row{&#1}%
3454     \expandafter\eql@transpose@scan@cell@next
3455     \fi
3456 }

```

**TODO:** describe

```

3457 \def\eql@transpose@output@row{%
3458     \ifnum\eql@row@<\eql@totalrows@
3459         \advance\eql@row@\@ne
3460         \eql@column@\z@
3461         \eql@transpose@output@col
3462         \ifnum\eql@row@<\eql@totalrows@
3463             \eql@scan@addto\\%
3464         \fi
3465     \expandafter\eql@transpose@output@row
3466     \fi
3467 }

```

**TODO:** describe

```
3468 \def\eql@transpose@output@col{%
3469   \ifnum\eql@column@<\eql@totalcolumns@
3470     \advance\eql@column@\@ne
3471     \csname eql@transpose@data@col@\the\eql@column@\endcsname
3472     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
3473     \ifnum\eql@column@<\eql@totalcolumns@
3474       \eql@scan@addto{\eql@transpose@complete}%
3475     \fi
3476     \expandafter\eql@transpose@output@col
3477   \fi
3478 }
```

### N.3 Measure

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

s@measure@line@begin

```
3479 \def\eql@columns@measure@line@begin{%
3480 (dev)\eql@dev{starting line \the\eql@row}%
3481   \global\eql@column@\z@
3482   \eql@numbering@measure@line@begin
3483   \eql@hook@linein
3484 }

3485 \def\eql@columns@measure@cell{%
3486   \eql@cellwidth@\wd\eql@cellbox@
3487   \ifdefined\eql@frame@cmd
3488     \eql@frame@measure
3489     \advance\eql@cellwidth@\eql@frame@margin@
3490   \fi
3491   \ifnum\eql@column@=\@ne
3492     \eql@widthdata@startrow
3493   \fi
3494   \ifodd\eql@column@
3495     \eql@shape@pos@\tw@
3496   \else
3497     \eql@shape@pos@\z@
3498   \fi
3499   \eql@shape@amount@\z@
3500   \eql@widthdata@savercell
3501   \ifodd\eql@column@\else
3502     \eql@widthdata@savesep
3503   \fi
3504   \kern\eql@cellwidth@
3505 }
```

mns@measure@line@end

```
3506 \def\eql@columns@measure@line@end{%
3507   \eql@punct@apply@line
3508   \eql@hook@lineout
3509   &\omit
3510   \ifnum\eql@column@>\eql@totalcolumns@
3511     \global\eql@totalcolumns@\eql@column@
3512   \fi
```



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

```

3513 \ifdefined\eql@frame@cmd
3514   \advance\eql@column@\@ne
3515   \wd\eql@cellbox@\z@
3516   \eql@columns@measure@cell
3517 \fi
3518 \eql@measure@tag
3519 }

```

`\eql@columns@measure`

```

3520 \def\eql@columns@measure{%
3521 (dev)\eql@dev@enter\eql@columns@measure
3522 \eql@totalcolumns@\z@
3523 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end

3524 \setbox\z@\vbox{\halign{%
3525   &%
3526   \global\advance\eql@column@\@ne
3527   \global\let\eql@cell@container\@empty
3528   \global\setbox\eql@cellbox@\hbox{%
3529     \@lign
3530     $\m@th\displaystyle
3531     \eql@hook@colin
3532     ##%
3533     \eql@class@innerleft
3534     \eql@hook@innerleft
3535     $%
3536   }%
3537   \eql@cell@container
3538   \hfil
3539   \eql@columns@measure@cell
3540   \global\let\eql@frame@prevcmd\eql@frame@cmd
3541   &%
3542   \eql@prevwidth@\wd\eql@cellbox@
3543   \let\eql@frame@cmd\eql@frame@prevcmd
3544   \global\advance\eql@column@\@ne
3545   \global\let\eql@cell@container\@empty
3546   \setbox\eql@cellbox@\hbox{%
3547     \@lign
3548     $\m@th\displaystyle
3549     \eql@hook@innerright
3550     \eql@class@innerright@sel
3551     ##%
3552     \eql@punct@apply@col
3553     \eql@hook@colout
3554     $%
3555   }%
3556   \eql@cell@container
3557   \eql@columns@measure@cell
3558   \hfil
3559 \crrc

3560 \noalign{%
3561   \eql@hook@blockbefore
3562 }%

```

```

3563 \eq@hook@blockin
3564 \eq@scan@body

3565 \ifvmode\else
3566 \eq@punct@apply@block
3567 \eq@hook@blockout
3568 \eq@display@endline
3569 \cr
3570 \fi
3571 \noalign{%
3572 \eq@hook@blockafter
3573 }%

```

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

```

3574 \omit
3575 \eq@column@\@ne
3576 \eq@columns@completerow
3577 \cr
3578 }}%

3579 \eq@measure@close

3580 \setbox\z@\vbox{%
3581 \unvbox\z@
3582 \unpenalty
3583 \global\setbox\@ne\lastbox
3584 }%
3585 \eq@totalwidth@\wd\@ne

```

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

```

3586 \let\eq@colwidth@tab\@empty
3587 \loop
3588 \setbox\@ne\hbox{%
3589 \unhbox\@ne
3590 \unskip
3591 \global\setbox\thr@\@ne\lastbox
3592 }%
3593 \ifhbox\thr@@
3594 \eq@colwidth@save{\wd\thr@@}%
3595 \repeat

3596 <dev>\eq@dev@leave\eq@columns@measure
3597 }

```

## N.4 Columns Placement

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

```

3598 \def\eq@columns@adjust{%
3599 \ifodd\eq@totalcolumns@
3600 \advance\eq@totalcolumns@\@ne
3601 \fi
3602 \eq@adjust@calc@columns
3603 }

```

## N.5 Print

**TODO:** describe

mns@print@line@begin

```
3604 \def\eq@columns@print@line@begin{%
3605 (dev)\eq@dev{starting line \the\eq@row}%
3606 \global\eq@column@z@
3607 \global\eq@line@pos@\eq@marginleft@
3608 \global\eq@line@width@z@
3609 \global\eq@line@avail@\eq@totalwidth@
3610 \global\eq@line@height@z@
3611 \global\eq@line@depth@z@
3612 \eq@numbering@print@line@begin
3613 \eq@hook@linein
3614 }
```

l@columns@print@cell

```
3615 \def\eq@columns@print@cell{%
3616 \eq@cellwidth@\wd\eq@cellbox@
3617 \ifodd\eq@column@
3618 \ifdefined\eq@frame@cmd
3619 \eq@frame@measure
3620 \advance\eq@cellwidth@\eq@frame@margin@
3621 \fi
3622 \dimen@z@
3623 \else
3624 \advance\eq@cellwidth@-\eq@prevwidth@
```

draw a frame

```
3625 \ifdefined\eq@frame@cmd
3626 \eq@frame@measure
3627 \advance\eq@cellwidth@\eq@frame@margin@
3628 \advance\eq@prevwidth@\eq@frame@margin@
3629 \eq@frame@print
3630 \fi
```

update height and depth

```
3631 \ifdim\ht\eq@cellbox@>\eq@line@height@
3632 \global\eq@line@height@\ht\eq@cellbox@
3633 \fi
3634 \ifdim\dp\eq@cellbox@>\eq@line@depth@
3635 \global\eq@line@depth@\dp\eq@cellbox@
3636 \fi
```

print box

```
3637 \kern-\eq@prevwidth@
3638 \unhbox\eq@cellbox@
3639 \dimen@-\eq@cellwidth@
3640 \fi
```

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

```
3641 \advance\dimen@\eq@colwidth@get\eq@column@\relax
3642 \kern\dimen@
```

update available and used space

```

3643 \dimen@eq\colwidth@get\eq\column@\relax
3644 \ifdim\eq\cellwidth@>\z@
3645   \ifdim\eq\line@width@=\z@
3646     \eq\line@avail@\eq\line@pos@
3647     \ifodd\eq\column@
3648       \advance\eq\line@avail@\dimen@
3649       \advance\eq\line@avail@-\eq\cellwidth@
3650     \fi
3651     \global\eq\line@avail@\eq\line@avail@
3652   \fi
3653   \eq\line@width@\eq\line@pos@
3654   \ifodd\eq\column@
3655     \advance\eq\line@width@\dimen@
3656   \else
3657     \advance\eq\line@width@\eq\cellwidth@
3658   \fi
3659   \global\eq\line@width@\eq\line@width@
3660 \fi
3661 \advance\eq\line@pos@\dimen@
3662 \ifodd\eq\column@\else
3663   \advance\eq\line@pos@\eq\colsep@
3664 \fi
3665 \global\eq\line@pos@\eq\line@pos@
3666 }

3667 \def\eq\columns@print@trailright{%
3668   &\omit
3669   \eq\prevwidth@\wd\eq\cellbox@
3670   \let\eq\frame@cmd\eq\frame@prevcmd
3671   \global\advance\eq\column@\@ne
3672   \eq\columns@print@cell
3673 }

```

lums@print@line@end

```

3674 \def\eq\columns@print@line@end{%
3675   \eq\punct@apply@line
3676   \eq\hook@lineout
3677 % \TODO add an even column with empty stuff if box processing deferred
3678   \ifodd\eq\column@
3679     \expandafter\eq\columns@print@trailright
3680   \fi
3681   \eq\columns@completerow
3682   \eq\columns@print@tag
3683 }

```

q\columns@print@tag

```

3684 \def\eq\columns@print@tag{%
3685   \dimen@\eq\totalwidth@
3686   \advance\dimen@\eq\colsep@
3687   \kern-\dimen@

```

determine first line available space

```

3688 \eq\display@firstavail@set\eq\line@avail@
3689 \eq\columns@overfull
3690 \eq\numbering@print@line@eval

```

```

3691 \if@eqnsw
3692   \eql@tagbox@make\eql@composetag@print
3693 \else
3694   \eql@raisetag@mode@z@
3695 \fi
3696 \eql@tagging@tagaddbox
3697 \eql@tagbox@print@cell
3698 }

```

\eql@columns@print

```

3699 \def\eql@columns@print{%
3700 (dev)\eql@dev@enter\eql@columns@print
3701   \eql@shape@align@disable
3702   \eql@display@halign@init\eql@columns@print@line@begin
3703   \eql@display@halign@letcr\eql@columns@print@line@end
3704   \tabskip\eql@marginleft@

3705   \halign{%
3706     &%
3707     \global\advance\eql@column@ \@ne
3708     \global\let\eql@cell@container\@empty
3709     \global\setbox\eql@cellbox@\hbox{%
3710       \eql@strut@cell
3711       \@lign
3712       $\m@th\displaystyle
3713       \eql@hook@colin
3714       ##%
3715       \eql@class@innerleft
3716       \eql@hook@innerleft
3717       \eql@tagging@mathsave
3718       $%
3719       \eql@tagging@mathaddlast
3720     }%
3721     \eql@cell@container
3722     \hfil
3723     \eql@columns@print@cell
3724     \global\let\eql@frame@prevcmd\eql@frame@cmd
3725     \tabskip\z@skip
3726     &%
3727     \eql@prevwidth@\wd\eql@cellbox@
3728     \let\eql@frame@cmd\eql@frame@prevcmd
3729     \global\advance\eql@column@ \@ne
3730     \global\let\eql@cell@container\@empty
3731     \setbox\eql@cellbox@\hbox{%
3732       \unhbox\eql@cellbox@
3733       \eql@strut@cell
3734       \@lign
3735       $\m@th\displaystyle
3736       \eql@hook@innerright
3737       \eql@class@innerright@sel
3738       ##%
3739       \eql@punct@apply@col
3740       \eql@hook@colout
3741       \eql@tagging@mathsave
3742       $%
3743       \eql@tagging@mathaddlast
3744     }%
3745     \eql@cell@container

```

```

3746      \eql@columns@print@cell
3747      \hfil
3748      \tabskip\eql@colsep@\relax
3749      \crcr

3750      \noalign{%
3751        \eql@display@halign@start
3752        \eql@numbering@print@block@begin
3753        \eql@hook@blockbefore
3754      }%
3755      \eql@hook@blockin
3756      \eql@scan@body
3757      \ifvmode\else
3758        \relax
3759        \eql@punct@apply@block
3760        \eql@hook@blockout
3761        \eql@display@endline
3762        \cr
3763      \fi
3764      \noalign{%
3765        \eql@hook@blockafter
3766        \eql@display@halign@end
3767      (dev)\eql@dev@leave\eql@columns@print
3768      }%
3769      \eql@tagging@tables@savealign
3770    }%
3771  }

```

## O Interface

### O.1 Scanning the Equation Body

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

#### Token Register.

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

```
3772 \newtoks\eql@scan@reg@
```

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

```

3773 \def\eql@scan@body@dump{\the\eql@scan@reg@}
3774 \def\eql@scan@body@rescan{%
3775   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
3776 \let\eql@scan@body\eql@scan@body@dump

```

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

```
3777 \long\def\eql@scan@addto#1{\eql@scan@reg@\expandafter{\the\eql@scan@reg@#1}}
```

**Environment Body.** The following mechanism scans the contents of an environment taking into account nested environments that may be contained in the body.

`\eql@scan@env` The macro `\eql@scan@env` starts the scan for the `\end{...}` command of the current environment. The argument is a call-back macro to process the body in `\eql@scan@reg@`:

```
3778 \def\eql@scan@env#1{%
3779 (dev)\eql@dev@enter\eql@scan@env
3780 \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenvir}}%
3781 \eql@scan@reg@{\def\eql@scan@stack{b}}%
```

We call `\eql@scan@env@iterate` which will scan until the next occurrence of `\end` and then count the number of occurrences of `\begin` before `\end` in `\eql@scan@stack`. If we simply called `\eql@scan@env@iterate` directly, the error message for an unwanted `\par` token (usually from a blank line) would refer to `\eql@scan@env@iterate` which would not be illuminating. We use a little finesse to get a more intelligible error message: We use the actual environment name as the name of the temporary function that is `\let` to `\eql@scan@env@iterate`:

```
3782 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
3783 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
3784 \ifdefined\eql@scan@par
3785 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
3786 \else
3787 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
3788 \fi
3789 \eql@scan@iterate
3790 }
```

`\eql@scan@env@iterate` `\eql@scan@env@iterate` takes two arguments: the first will consist of all text up to the next `\end` command, the second will be the `\end` command's argument. If there are any extra `\begin` commands in the body text, a marker is pushed onto a stack via `\eql@scan@env@count`. An empty state for this stack means that we have reached the `\end` that matches our original `\begin`. Otherwise we need to include the `\end` and its argument in the material that we are adding to our environment body accumulator:

```
3791 \long\def\eql@scan@env@iterate#1\end#2{%
3792 \edef\eql@scan@stack{%
3793 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
3794 \ifx\@empty\eql@scan@stack
3795 \@checkend{#2}%
3796 \eql@scan@addto{#1}%
3797 \expandafter\let\eql@scan@iterate\eql@scan@env@org
3798 (dev)\eql@dev@leave\eql@scan@env
3799 \expandafter\eql@scan@end
3800 \else
3801 \eql@scan@addto{#1\end{#2}}%
3802 \expandafter\eql@scan@iterate
3803 \fi
3804 }
```

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

```
3805 \def\eql@scan@env@iterate@nopar#1\end#2{\eql@scan@env@iterate#1\end{#2}}
```

`\eql@scan@env@count` When adding a piece of the current environment's contents to `\eql@scan@reg@`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

3806 \long\def\eq1@scan@env@count#1\begin#2{%
3807   \ifx\end#2\else b\expandafter\eq1@scan@env@count\fi
3808 }

```

The call-back macro `\eq1@scan@env@cancel` ignores the body as well as the end clause for the environment:

```

3809 \def\eq1@scan@env@cancel{%
3810   \@namedef{end\@currentvir}{\ignorespacesafterend}%
3811 }

```

**Square Brackets.** The following is a version of the above mechanism that scans for an equation body enclosed by `\[...]` paying attention to potential further instances of the square bracket enclosures contained in the body.

`\eq1@scan@sqr` Start scanning for `\[`:

```

3812 \def\eq1@scan@sqr#1{%
3813   (dev)\eq1@dev@enter\eq1@scan@sqr
3814   \def\eq1@scan@end{#1\]}%
3815   \eq1@scan@reg@{\}\def\eq1@scan@stack{b}%
3816   \let\eq1@scan@sqr@org\[%\]
3817   \ifdefined\eq1@scan@par
3818     \let\[\eq1@scan@sqr@iterate%\]
3819   \else
3820     \let\[\eq1@scan@sqr@iterate@nopar%\]
3821   \fi
3822   \[%\]
3823 }

```

`\eq1@scan@sqr@iterate` Iterate until we find a balanced pairing of square brackets. Then call the call-back macro:

```

3824 \long\def\eq1@scan@sqr@iterate#1\{%
3825   \edef\eq1@scan@stack{%
3826     \eq1@scan@sqr@count#1\[\]\expandafter\@gobble\eq1@scan@stack}%
3827   \ifx\@empty\eq1@scan@stack
3828     \let\[\eq1@scan@sqr@org%\]
3829     \eq1@scan@addto{#1}%
3830   (dev)\eq1@dev@leave\eq1@scan@sqr
3831     \expandafter\eq1@scan@end
3832   \else
3833     \eq1@scan@addto{#1\]}%
3834     \expandafter\[%\]
3835   \fi
3836 }

```

`\eq1@scan@sqr@iterate@nopar` Version of `\eq1@scan@sqr@iterate` which does not accept `\par` within the argument:

```

3837 \def\eq1@scan@sqr@iterate@nopar#1\{\eq1@scan@sqr@iterate#1\}

```

`\eq1@scan@sqr@count` Push a ‘b’ for every encountered instance of ‘`\[`’:

```

3838 \long\def\eq1@scan@sqr@count#1\[#2{%\]
3839   \ifx\]#2\else b\expandafter\eq1@scan@sqr@count\fi
3840 }

```

`\eq1@scan@sqrang@cancel` The call-back macro `\eq1@scan@sqrang@cancel` ignores the body and the closing bracket:

```

3841 \def\eq1@scan@sqrang@cancel{\expandafter\ignorespaces\@gobble}

```



**Angle Brackets.** The following is another version of the mechanism which scans for an equation body enclosed by `\<... \>`.

`\eql@scan@ang` Start scanning for `\>`:

```

3842 \def\eql@scan@ang#1{%
3843 (dev)\eql@dev@enter\eql@scan@ang
3844 \def\eql@scan@end{#1\>}%
3845 \eql@scan@reg@{ }\def\eql@scan@stack{b}%
3846 \let\eql@scan@ang@org\<%\>
3847 \ifdefined\eql@scan@par
3848 \let\<\eql@scan@ang@iterate%\>
3849 \else
3850 \let\<\eql@scan@ang@iterate@nopar%\>
3851 \fi
3852 \<%\>
3853 }
```

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

```

3854 \long\def\eql@scan@ang@iterate#1\>{%
3855 \edef\eql@scan@stack{%
3856 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
3857 \ifx\@empty\eql@scan@stack
3858 \let\<\eql@scan@ang@org%\>
3859 \eql@scan@addto{#1}%
3860 (dev)\eql@dev@leave\eql@scan@ang
3861 \expandafter\eql@scan@end
3862 \else
3863 \eql@scan@addto{#1\>}%
3864 \expandafter\<%\>
3865 \fi
3866 }
```

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

```

3867 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}
```

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

```

3868 \long\def\eql@scan@ang@count#1\<#2{%\>
3869 \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
3870 }
```

## O.2 Options Processing

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

```

3871 \def\eql@equations@testall{\eql@parseopt\eql@equations@parseopt}
3872 \def\eql@equations@parseopt{%
3873 \ifx\eql@parseopt@token*%
3874 \let\eql@parseopt@next\eql@parseopt@nonumber
3875 \fi
3876 \ifx\eql@parseopt@token!%
3877 \let\eql@parseopt@next\eql@parseopt@donumber
```

```

3878 \fi
3879 % \ifx\eql@parseopt@token~%
3880 % \let\eql@parseopt@next\eql@parseopt@lines
3881 % \fi
3882 \ifx\eql@parseopt@token/%
3883 \let\eql@parseopt@next\eql@parseopt@transpose
3884 \fi
3885 \ifx\eql@parseopt@token[%]
3886 \let\eql@parseopt@next\eql@parseopt@opt
3887 \fi
3888 \ifx\eql@parseopt@token\eql@atxi
3889 \let\eql@parseopt@next\eql@parseopt@label
3890 \fi
3891 \ifx\eql@parseopt@token\eql@atxii
3892 \let\eql@parseopt@next\eql@parseopt@label
3893 \fi
3894 \ifx\eql@parseopt@token\label
3895 \let\eql@parseopt@next\eql@parseopt@end
3896 \fi
3897 \ifx\eql@parseopt@token.%
3898 \let\eql@parseopt@next\eql@parseopt@punctdot
3899 \fi
3900 \ifx\eql@parseopt@token,%
3901 \let\eql@parseopt@next\eql@parseopt@punctcomma
3902 \fi
3903 \ifx\eql@parseopt@token~%
3904 \let\eql@parseopt@next\eql@parseopt@punctoff
3905 \fi
3906 \ifx\eql@parseopt@token-%
3907 \let\eql@parseopt@next\eql@parseopt@single
3908 \fi
3909 \ifx\eql@parseopt@token=%
3910 \let\eql@parseopt@next\eql@parseopt@lines
3911 \fi
3912 \ifx\eql@parseopt@token|%
3913 \let\eql@parseopt@next\eql@parseopt@columns
3914 \fi
3915 }

```

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

```

3916 \def\eql@equations@processopt{%
3917 \let\eql@tags@container@block\eql@tags@container@clear
3918 \let\eql@tags@frame@cmd\@firstofone
3919 \let\eql@skip@force@above\@undefined
3920 \let\eql@skip@force@below\@undefined
3921 \let\eql@skip@force@leave\@undefined
3922 \let\eql@display@linewidth\@undefined
3923 \let\eql@display@marginleft\@undefined
3924 \let\eql@display@marginright\@undefined
3925 \eql@abovespace@\z@skip
3926 \eql@belowspace@\z@skip
3927 \eql@displaybreak@prepen@\@MM
3928 \eql@displaybreak@postpen@\@MM
3929 \eql@nextopt@process{equations}%

```

```

3930 \let\eql@punct@block\eql@punct@main
3931 \let\eql@punct@main\relax
3932 \eql@indent@\glueexpr\eql@indent@val\relax
3933 \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
3934 }

```

### O.3 Single-Line Main

In the following, we define the main routine for the single-line equation mode.

`\eql@single@cr` Cannot use line breaks, produce an error message:

```

3935 \def\eql@single@cr{%
3936 \eql@error{Cannot use '\string\\' within display equation.
3937 Please switch to equations environment}}%
3938 }

```

`\eql@single@start` Opening code for single-line equation. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, set native vs. manual equation tag mode, install error message for using `\.`. Hand over to mode-specific opening:

```

3939 \def\eql@single@start{%
3940 \eql@display@enter
3941 \eql@tagging@start
3942 \eql@dollar@begin
3943 \eql@display@adjust
3944 \eql@numbering@init
3945 \eql@stack@save@equations
3946 \eql@numbering@single@init
3947 \ifdefined\eql@single@crerror\else
3948 \let\\ \eql@single@cr
3949 \fi
3950 \ifdefined\eql@single@native
3951 \let\eql@single@start@sel\eql@single@start@native
3952 \let\eql@single@end@sel\eql@single@end@native
3953 \else
3954 \let\eql@single@start@sel\eql@single@start@print
3955 \let\eql@single@end@sel\eql@single@end@print
3956 \fi
3957 \eql@single@start@sel
3958 }

```

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

```

3959 \def\eql@single@end{%
3960 \eql@punct@apply@block
3961 \eql@hook@eqout
3962 \eql@single@end@sel
3963 \eql@stack@restore
3964 \eql@dollar@end
3965 \eql@tagging@end
3966 \eql@display@leave
3967 }

```

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

```
3968 \def\eql@single@main{%
3969   \expandafter\eql@single@start
3970   \eql@scan@body
3971   \eql@single@end
3972 }
```

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

```
3973 \def\eql@mode@single{%
3974   \ifdefined\eql@single@dscan
3975     \let\eql@equations@main\eql@single@main
3976     \let\eql@equations@end\@empty
3977   \else
3978     \let\eql@equations@main\@undefined
3979     \let\eql@equations@end\eql@single@end
3980   \fi
3981 }
```

## O.4 Multi-Line Main

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

```
3982 \let\eql@multi@mode@lines\eql@false
```

`\eql@multi@main` Main routine for multi-line modes. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, initialise macros for use within equations: **TODO:** shove depends on lines vs columns

```
3983 \def\eql@multi@main{%
3984   \eql@display@enter
3985   \eql@tagging@start
3986   \eql@dollar@dollar@begin
3987   \eql@display@adjust
3988   \eql@numbering@init
3989   \eql@stack@save@equations
3990   \ifdefined\eql@transpose@active
3991     \ifdefined\eql@multi@mode@lines\else
3992       \eql@transpose
3993     \fi
3994   \fi
3995   \ifdefined\eql@numbering@subeq@use
3996     \eql@numbering@subeq@init
3997   \fi
3998   \eql@display@init
3999   \let\intertext\eql@intertext
4000   \let\endintertext\endeql@intertext
4001   \eql@shape@align@enable
```

Now measure the given multi-line equations body:

```
4002   \ifdefined\eql@multi@mode@lines
4003     \eql@lines@measure
4004   \else
4005     \ifdefined\eql@ampproof@active
4006       \eql@ampproof
```

```

4007   \fi
4008   \eql@columns@measure
4009 \fi

```

If only a single equation number is used for subequation numbering, revert to normal equation numbering. If only a single column is used in columns mode, may fallback to lines mode. Switching from columns to lines mode, the width can be incorrect, expect only minor discrepancies, but for accurateness, should call `\eql@lines@measure`:

```

4010 \ifx\eql@numbering@subeq@use\@ne
4011   \eql@numbering@subeq@revert
4012 \fi
4013 \ifdefined\eql@multi@mode@lines\else
4014   \ifdefined\eql@multi@lines@fallback
4015     \ifnum\eql@totalcolumns@=\@ne
4016       \let\eql@multi@mode@lines\eql@true
4017       \ifx\eql@multi@lines@fallback\z@\else
4018         \eql@lines@measure
4019       \fi
4020     \fi
4021   \fi
4022 \fi

```

Adjust the multi-line equations body:

```

4023 \ifdefined\eql@multi@mode@lines
4024   \eql@lines@adjust
4025 \else
4026   \eql@columns@adjust
4027 \fi

```

Now print the multi-line equations body:

```

4028 \eql@display@print
4029 \eql@numbering@print@init
4030 \ifdefined\eql@multi@mode@lines
4031   \eql@lines@print
4032 \else
4033   \eql@columns@print
4034 \fi
4035 \eql@display@close

```

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

```

4036 \ifdefined\eql@numbering@subeq@use
4037   \eql@numbering@subeq@close
4038 \fi
4039 \eql@stack@restore
4040 \eql@dollar@end
4041 \eql@tagging@end
4042 \eql@display@leave
4043 }

```

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

```

\eql@mode@lines
4044 \def\eql@mode@columns{%
4045   \let\eqlequations@main\eql@multi@main
4046   \let\eqlequations@end\@empty
4047   \let\eql@multi@mode@lines\eql@false
4048 }

```

```

4049 \def\eql@mode@lines{%
4050   \let\eql@equations@main\eql@multi@main
4051   \let\eql@equations@end\@empty
4052   \let\eql@multi@mode@lines\eql@true
4053 }

```

## O.5 Equations Environment

We now declare the main environment and its symbolic versions.

### Environment.

**equations** (*env.*) Declare the main equations environment. If already in math mode, fail and cancel the environment body. Otherwise scan for optional arguments and pass on to `\eql@equations@start`:

```

4054 \newenvironment{equations}{%
4055 (dev)\eql@dev@enterenv
4056   \ifmmode
4057     \eqLError@mathmode{\string\begin{\@currenvir}}%
4058     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4059   \else
4060     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4061     \expandafter\eql@equations@start
4062   \fi
4063 }{%
4064   \eql@equations@end
4065   \ignorespacesafterend
4066 (dev)\eql@dev@leaveenv
4067 }

```

`\eql@equations@start` The macro `\eql@equations@start` first processes the arguments. Depending on the chosen mode of operation, scan the environment body passing on to `\eql@equations@main` or process a single-line equation via `\eql@single@start`:

```

4068 \def\eql@equations@start{%
4069   \eql@equations@processopt
4070   \ifdefined\eql@equations@main
4071     \expandafter\eql@scan@env\expandafter\eql@equations@main
4072   \else
4073     \expandafter\eql@single@start
4074   \fi
4075 }

```

### Square Brackets.

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

```

4076 \newenvironment{equations@sqr}{}{}

```

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

```

4077 \protected\def\eql@equations@sqr@open{%

```

```

4078 \ifmmode
4079   \eqLError@mathmode{\string\[\dots\string\]}%
4080   \expandafter\eqL@scan@sqr\expandafter\eqL@scan@sqrang@cancel
4081 \else
4082 (dev)\eqL@dev@enter{\[\dots\string\]}%
4083   \expandafter\eqnaddopt\expandafter{\eqL@equations@sqr@opt}%
4084   \begin{equations@sqr}%
4085   \let\]\eqL@equations@sqr@close
4086   \expandafter\eqL@ampprotect\expandafter\eqL@equations@testall
4087   \expandafter\eqL@equations@sqr@start
4088 \fi
4089 }

```

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

```

4090 \def\eqL@equations@sqr@start{%
4091   \eqL@equations@processopt
4092   \ifdefined\eqL@equations@main
4093     \expandafter\eqL@scan@sqr\expandafter\eqL@equations@main
4094   \else
4095     \expandafter\eqL@single@start
4096   \fi
4097 }

```

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

```

4098 \protected\def\eqL@equations@sqr@close{%
4099   \eqL@equations@end
4100 (dev)\eqL@dev@leave{\[\dots\string\]}%
4101   \end{equations@sqr}%
4102   \ignorespaces
4103 }

```

**TODO:** describe

```

\eqL@sqr@open
\eqL@sqr@close
4104 \let\eqL@sqr@open\eqL@equations@sqr@open
4105 \protected\def\eqL@sqr@close{%
4106   \eqLError{'\string\]' may only close '\string\[']}%\]
4107 }

```

## Angle Brackets.

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

```

4108 \newenvironment{equations@ang}{}{}
4109 \newenvironment{equationsbox@ang}{}{}

```

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

```

4110 \protected\def\eqL@ang@open{%
4111 (dev)\eqL@dev@enter{\<\dots\string\>}%
4112   \ifmmode
4113     \expandafter\eqnaddopt\expandafter{\eqL@box@ang@opt}%
4114     \begin{equationsbox@ang}%
4115     \let\>\eqL@box@ang@close
4116     \expandafter\eqL@ampprotect\expandafter\eqL@box@testall

```

```

4117     \expandafter\eql@box@start
4118 \else
4119     \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
4120     \begin{equations@ang}%
4121     \let\>\eql@equations@ang@close
4122     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4123     \expandafter\eql@equations@ang@start
4124 \fi
4125 }

```

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

```

4126 \protected\def\eql@ang@close{%
4127   \eql@error{'\string\>' may only close '\string\<'}%\>
4128 }

```

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

```

4129 \def\eql@equations@ang@start{%
4130   \eql@equations@processopt
4131   \ifdefined\eql@equations@main
4132     \expandafter\eql@scan@ang\expandafter\eql@equations@main
4133   \else
4134     \expandafter\eql@single@start
4135   \fi
4136 }

```

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

```

4137 \def\eql@equations@ang@close{%
4138   \eql@equations@end
4139   \end{equations@ang}%
4140 (dev)\eql@dev@leave{\<...\string\>}%
4141   \ignorespaces
4142 }

```

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

```

4143 \def\eql@box@ang@close{%
4144   \eql@box@end
4145   \end{equationsbox@ang}%
4146 (dev)\eql@dev@leave{\<...\string\>}%
4147   \ignorespaces
4148 }

```

## P Options

### P.1 Selection Tools

`\eql@decide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```

4149 \def\eql@decide@abovebelow#1#2#3#4#5{%
4150   \eql@decide@select{#1}{#2}{#3}{%
4151     {,abovebelow,both,tb}{#4#5},%
4152     {above,top,t}{#4},%
4153     {below,bottom,b}{#5}}%

```



`eq1@decide@situation` Select a particular vertical spacing situation and store it in the macro #4:

```

4154 \def\eq1@decide@situation#1#2#3#4{%
4155   \eq1@decide@select{#1}{#2}{#3}{%
4156     {{long}{\def#4{0}}},%
4157     {{short}{\def#4{1}}},%
4158     {{cont}{\def#4{2}}},%
4159     {{par}{\def#4{3}}},%
4160     {{top}{\def#4{4}}},%
4161     {{noskip}{\def#4{5}}},%
4162     {{medskip}{\def#4{6}}}}}
```

## P.2 Options Declarations

We now declare all key-value pairs for options sorted by their category.

**Modes for Equations Box Environment.** Declare horizontal and vertical alignment modes for the boxed equations environment. Also declare spacing of columns:

```

4163 \eq1@define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{%
4164   \eq1@mode@stacked}
4165 \eq1@define@key{equationsbox}{aligned,align,al,columns,col}[]{%
4166   \eq1@mode@aligned}
4167 \eq1@define@key{equationsbox}{top,t}[]{\let\eq1@box@box\vtop}
4168 \eq1@define@key{equationsbox}{center,c}[]{\let\eq1@box@box\vcenter}
4169 \eq1@define@key{equationsbox}{bottom,b}[]{\let\eq1@box@box\vbox}
4170 \eq1@define@key{setup}{boxangopt}[]{%
4171   \def\eq1@box@ang@opt{columns,#1}}
```

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

```

4172 \eq1@define@key{equations}{equation,eq,single,1}[]{\eq1@mode@single}
4173 \eq1@define@key{equations}{gathered,gather,ga,lines,ln}[]{%
4174   \eq1@mode@lines}
4175 \eq1@define@key{equations}{aligned,align,al,columns,col}[]{%
4176   \eq1@mode@columns}
4177 \eq1@define@key{equations,setup}{transpose}[true]{%
4178   \eq1@decide@select{#3}{#2}{#1}{%
4179     {\eq1@decide@false{\let\eq1@transpose@active\eq1@false}},%
4180     {{noamp,plain,restricted}{\let\eq1@transpose@active\eq1@true}},%
4181     {{\eq1@decide@true,amp,cont}{\let\eq1@transpose@active=+}}}}
4182 \eq1@define@key{equations}{native}[true]{%
4183   \eq1@decide@bool{#3}{#2}{#1}\eq1@single@native%
4184   \ifdefined\eq1@single@native\let\eq1@layoutright\eq1@false\fi}
4185 \eq1@define@key{setup}{native}[true]{%
4186   \eq1@decide@bool{#3}{#2}{#1}\eq1@single@native}
4187 \eq1@define@key{setup}{scanequation}[true]{%
4188   \eq1@decide@bool{#3}{#2}{#1}\eq1@single@doscans}
4189 \eq1@define@key{setup}{sqropt}[]{%
4190   \def\eq1@equations@sqr@opt{equation,#1}}
4191 \eq1@define@key{setup}{angopt}[]{%
4192   \def\eq1@equations@ang@opt{columns,#1}}
```

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

```

4193 \def\eql@keycat{equations,equationsbox,setup}
4194 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
4195 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
4196   {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
4197   {{cell}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
4198   {{tag}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
4199   {\eql@decide@true
4200     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
4201 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings concerning page breaks:

```

4202 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4203   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
4204   {{high,3}{\eql@displaybreak@pre3}},%
4205   {{med,medium,2}{\eql@displaybreak@pre2}},%
4206   {{low,1}{\eql@displaybreak@pre1}},%
4207   {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
4208   {{default,inherit,-1}{\eql@displaybreak@pre\m@ne}}}}
4209 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4210   {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
4211   {{high,3}{\eql@displaybreak@post3}},%
4212   {{med,medium,2}{\eql@displaybreak@post2}},%
4213   {{low,1}{\eql@displaybreak@post1}},%
4214   {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
4215   {{default,inherit,-1}{\eql@displaybreak@post\m@ne}}}}
4216 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4217   \eql@decide@select{#3}{#2}{#1}{%
4218     {{full,4}{\eql@displaybreak@inter4}},%
4219     {{high,3}{\eql@displaybreak@inter3}},%
4220     {{med,medium,2}{\eql@displaybreak@inter2}},%
4221     {{low,1}{\eql@displaybreak@inter1}},%
4222     {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
4223 \eql@define@key{equations}{prepenalty}{%
4224   \eql@displaybreak@prepen@\numexpr#1\relax}
4225 \eql@define@key{equations}{postpenalty}{%
4226   \eql@displaybreak@postpen@\numexpr#1\relax}
4227 \eql@define@key{equations,setup}{interpenalty}{%
4228   \interdisplaylinepenalty\numexpr#1\relax}

```

**TODO:** describe **TODO:** vspace! for fixed glue after break

```

4229 \eql@define@key{control}{vspace}[]{\eql@vspace@set{#1}}
4230 \eql@define@key{control}{vspace*}[]{\eql@vspace@setfixed\eql@vspace@set{#1}}
4231 \eql@define@key{control}{break}[4]{\eql@displaybreak@level[#{#1}]}
4232 \eql@define@key{control}{penalty}[]{\eql@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4233 \eql@define@key\eql@keycat{displayheight}[strut]{%
4234   \eql@decide@select{#3}{#2}{#1}{%
4235     {\eql@decide@false{\let\eql@display@height\@undefined}},%
4236     {{strut}{\def\eql@display@height{\ht\eql@strutbox@}}},%
4237     {\relax{\def\eql@display@height{#1}}}}
4238 \eql@define@key\eql@keycat{displaydepth}[strut]{%
4239   \eql@decide@select{#3}{#2}{#1}{%
4240     {\eql@decide@false{\let\eql@display@depth\@undefined}},%
4241     {{strut}{\def\eql@display@depth{\dp\eql@strutbox@}}},%
4242     {\relax{\def\eql@display@depth{#1}}}}

```

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

as short would apply...

```

4243 \eql@define@key{equations}{noskip}[]{%
4244   \eql@decide@abovebelow{#3}{#2}{#1}%
4245   {\def\eql@skip@force@above{5}}%
4246   {\def\eql@skip@force@below{5}}}
4247 \eql@define@key{equations}{short}[above]{%
4248   \eql@decide@abovebelow{#3}{#2}{#1}%
4249   {\def\eql@skip@force@above{1}}%
4250   {\def\eql@skip@force@below{1}}}
4251 \eql@define@key{equations}{long}[]{%
4252   \eql@decide@abovebelow{#3}{#2}{#1}%
4253   {\def\eql@skip@force@above{0}}%
4254   {\def\eql@skip@force@below{0}}}
4255 \eql@define@key{equations}{medskip}[]{%
4256   \eql@decide@abovebelow{#3}{#2}{#1}%
4257   {\def\eql@skip@force@above{6}}%
4258   {\def\eql@skip@force@below{6}}}
4259 \eql@define@key{equations}{par}[par]{%
4260   \eql@decide@select{#3}{#2}{#1}{%
4261     {{default,}\let\eql@skip@force@leave\undefined}},%
4262     {{cont,hmode}\let\eql@skip@force@leave\z@}},%
4263     {{par,vmode}\let\eql@skip@force@leave\@ne
4264       \ifdefined\eql@skip@force@below\else
4265         \def\eql@skip@force@below{3}%
4266       \fi}},%
4267     {{top}\let\eql@skip@force@leave\tw@
4268       \ifdefined\eql@skip@force@below\else
4269         \def\eql@skip@force@below{4}
4270       \fi}}}}

```

Specify vertical spacing explicitly:

```

4271 \eql@define@key{equations}{skip}{%
4272   \def\eql@skip@force@above{7}%
4273   \def\eql@skip@custom@above{#1}%
4274   \let\eql@skip@force@below\eql@skip@force@above
4275   \let\eql@skip@custom@below\eql@skip@custom@above}
4276 \eql@define@key{equations}{aboveskip}{%
4277   \def\eql@skip@force@above{7}%
4278   \def\eql@skip@custom@above{#1}}
4279 \eql@define@key{equations}{belowskip}{%
4280   \def\eql@skip@force@below{7}%
4281   \def\eql@skip@custom@below{#1}}
4282 \eql@define@key{equations}{abovespace}{%
4283   \advance\eql@abovespace@glueexpr#1\relax}
4284 \eql@define@key{equations}{belowspace}{%
4285   \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4286 \eql@define@key{intertext}{skip}{%
4287   \def\eql@skip@force@above{7}%
4288   \def\eql@skip@custom@above{#1}%
4289   \let\eql@skip@force@below\eql@skip@force@above
4290   \let\eql@skip@custom@below\eql@skip@custom@above}
4291 \eql@define@key{intertext}{aboveskip}{%
4292   \def\eql@skip@force@below{7}%
4293   \def\eql@skip@custom@below{#1}}
4294 \eql@define@key{intertext}{belowskip}{%

```

```

4295 \def\eql@skip@force@above{7}%
4296 \def\eql@skip@custom@above{#1}}
4297 \eql@define@key{intertext}{noskip}[]{%
4298 \eql@decide@abovebelow{#3}{#2}{#1}%
4299 {\def\eql@skip@force@below{5}}%
4300 {\def\eql@skip@force@above{5}}}}
4301 \eql@define@key{intertext}{short}[]{%
4302 \eql@decide@abovebelow{#3}{#2}{#1}%
4303 {\def\eql@skip@force@below{1}}%
4304 {\def\eql@skip@force@above{1}}}}
4305 \eql@define@key{intertext}{long}[]{%
4306 \eql@decide@abovebelow{#3}{#2}{#1}%
4307 {\def\eql@skip@force@below{0}}%
4308 {\def\eql@skip@force@above{0}}}}
4309 \eql@define@key{intertext}{medskip}[]{%
4310 \eql@decide@abovebelow{#3}{#2}{#1}%
4311 {\def\eql@skip@force@below{6}}%
4312 {\def\eql@skip@force@above{6}}}}

```

Configure general vertical spacing behaviour for various situations:

```

4313 \eql@define@key{setup}{skip,longskip}{%
4314 \abovedisplayskip\glueexpr#1\relax
4315 \belowdisplayskip\abovedisplayskip
4316 \def\eql@skip@long@above{#1}%
4317 \let\eql@skip@long@below\eql@skip@long@above}
4318 \eql@define@key{setup}{aboveskip,abovelongskip}{%
4319 \abovedisplayskip\glueexpr#1\relax
4320 \def\eql@skip@long@above{#1}}
4321 \eql@define@key{setup}{belowskip,belowlongskip}{%
4322 \belowdisplayskip\glueexpr#1\relax
4323 \def\eql@skip@long@below{#1}}
4324 \eql@define@key{setup}{aboveshortskip}{%
4325 \abovedisplayshortskip\glueexpr#1\relax
4326 \def\eql@skip@short@above{#1}}
4327 \eql@define@key{setup}{belowshortskip}{%
4328 \belowdisplayshortskip\glueexpr#1\relax
4329 \def\eql@skip@short@below{#1}}
4330 \eql@define@key{setup}{tagskip}{%
4331 \def\eql@skip@tag@above{#1}%
4332 \let\eql@skip@tag@below\eql@skip@tag@above}
4333 \eql@define@key{setup}{abovetagskip}{%
4334 \def\eql@skip@tag@above{#1}}
4335 \eql@define@key{setup}{belowtagskip}{%
4336 \def\eql@skip@tag@below{#1}}
4337 \eql@define@key{setup}{medskip}{%
4338 \def\eql@skip@med@above{#1}%
4339 \let\eql@skip@med@below\eql@skip@med@above}
4340 \eql@define@key{setup}{abovemedskip}{%
4341 \def\eql@skip@med@above{#1}}
4342 \eql@define@key{setup}{belowmedskip}{%
4343 \def\eql@skip@med@below{#1}}
4344 \eql@define@key{setup}{medtagskip}{%
4345 \def\eql@skip@medtag@above{#1}%
4346 \let\eql@skip@medtag@below\eql@skip@medtag@above}
4347 \eql@define@key{setup}{abovemedtagskip}{%
4348 \def\eql@skip@medtag@above{#1}}
4349 \eql@define@key{setup}{belowmedtagskip}{%
4350 \def\eql@skip@medtag@below{#1}}

```

```

4351 \eqld@define@key{setup}{abovetopskip}{%
4352   \def\eqld@skip@top@above{#1}}
4353 \eqld@define@key{setup}{belowtopskip}{%
4354   \def\eqld@skip@top@below{#1}}
4355 \eqld@define@key{setup}{aboveparskip}{%
4356   \def\eqld@skip@par@above{#1}}
4357 \eqld@define@key{setup}{belowparskip}{%
4358   \def\eqld@skip@par@below{#1}}
4359 \eqld@define@key{setup}{abovepartagskip}{%
4360   \def\eqld@skip@partag@above{#1}}
4361 \eqld@define@key{setup}{belowpartagskip}{%
4362   \def\eqld@skip@partag@below{#1}}
4363 \eqld@define@key{setup}{abovecontskip}{%
4364   \eqld@decide@select{#3}{#2}{#1}{%
4365     {\hide}{\def\eqld@skip@cont@above{\eqld@spread@val-\eqld@skip@long@below}}},%
4366     {\relax{\def\eqld@skip@cont@above{#1}}}}}%
4367 \eqld@define@key{setup}{belowcontskip}{%
4368   \def\eqld@skip@cont@below{#1}}
4369 \eqld@define@key{setup}{shortmode}{%
4370   \eqld@decide@select{#3}{#2}{#1}{%
4371     {\off,never,no}{\def\eqld@skip@mode@short{0}}},%
4372     {\above,neverbelow,notbelow,belowoff}{\def\eqld@skip@mode@short{1}}},%
4373     {\belowone,belowsingle}{\def\eqld@skip@mode@short{2}}},%
4374     {\belowall,always,on}{\def\eqld@skip@mode@short{3}}}}}%
4375 \eqld@define@key{setup}{abovecontmode}{%
4376   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@cont@above}%
4377 \eqld@define@key{setup}{belowcontmode}{%
4378   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@cont@below}%
4379 \eqld@define@key{setup}{aboveparmode}{%
4380   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@par@above}%
4381 \eqld@define@key{setup}{belowparmode}{%
4382   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@par@below}%
4383 \eqld@define@key{setup}{abovetopmode}{%
4384   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@above}%
4385 \eqld@define@key{setup}{belowtopmode}{%
4386   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@below}%

```

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

```

4387 \def\eqld@keycat{equations,subequations}
4388 \eqld@define@key\eqld@keycat{label}{\eqld@tags@addblock@label{#1}}
4389 \eqld@define@key\eqld@keycat{labelname}{\eqld@tags@addblock@name{#1}}
4390 \eqld@define@key\eqld@keycat{tag}{\eqld@tags@addblock@tag{#1}}
4391 \eqld@define@key\eqld@keycat{tag*}{%
4392   \eqld@tags@addblock@tagform@off\eqld@tags@addblock@tag{#1}}
4393 \eqld@define@key\eqld@keycat{taglabel}{\eqld@tags@addblock@ref{#1}}

```

**TODO:** describe

```

4394 \eqld@define@key{control}{label}{\eqld@tags@add@label{#1}}
4395 \eqld@define@key{control}{labelname}{\eqld@tags@add@name{#1}}
4396 \eqld@define@key{control}{tag}{\eqld@tags@add@tag{#1}}
4397 \eqld@define@key{control}{tag*}{\eqld@tags@add@tagform@off\eqld@tags@add@tag{#1}}
4398 \eqld@define@key{control}{taglabel}{\eqld@tags@add@ref{#1}}
4399 \eqld@define@key{control}{raisetag}{\eqld@tags@add@raiseamount{#1}}
4400 \eqld@define@key{control}{raisetag!}[]{\eqld@tags@add@raisemode\@ne}

```

**TODO:** describe

```

4401 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
4402 \eqld@define@key{setup}{autolabel}[true]{%
4403   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}
4404 \eqld@define@key{setup}{autotag}[true]{%
4405   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}

```

**Tag Spacing.** Configure horizontal spacing for equation tags:

```

4406 \def\eqld@keycat{equations,setup}
4407 \eqld@define@key\eqld@keycat{tagmargin}[auto]{%
4408   \eqld@decide@select{#3}{#2}{#1}{%
4409     {auto,\eqld@decide@false}{\let\eqld@tagmargin@val\undefined}},%
4410     {\relax{\def\eqld@tagmargin@val{#1}}}}}%
4411 \eqld@define@key\eqld@keycat{tagmargin*}{%
4412   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
4413 \eqld@define@key\eqld@keycat{tagmarginratio}{%
4414   \eqld@tagmargin@ratio@\dimexpr#1pt\relax}
4415 \eqld@define@key\eqld@keycat{tagmarginthreshold}{%
4416   \def\eqld@tagmargin@threshold{#1}}
4417 \eqld@define@key\eqld@keycat{mintagsep}{\def\eqld@tagsepmin@val{#1}}
4418 \eqld@define@key\eqld@keycat{mintagwidth}{%
4419   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
4420 \eqld@define@key\eqld@keycat{mintagwidth*}{\settowidth\eqld@tagwidthmin@{#1}}

```

**Tag Layout.** Configure methods to declare equation tag layout:

```

4421 \def\eqld@keycat{equations,setup}
4422 \eqld@define@key\eqld@keycat{tagbox,taglayout}{%
4423   \eqld@tags@taglayout@set{#1}}
4424 \eqld@define@key\eqld@keycat{tagbox*,taglayout*}{%
4425   \eqld@tags@taglayout@set@direct{#1}}
4426 \eqld@define@key\eqld@keycat{tagform}{%
4427   \eqld@tags@tagform@set{#1}}
4428 \eqld@define@key\eqld@keycat{tagform*}{%
4429   \eqld@tags@tagform@set@direct{#1}}
4430 \eqld@define@key\eqld@keycat{subeqtemplate}{%
4431   \def\eqld@subequations@template####1####2{#1}%
4432   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}}
4433 \eqld@define@key{control}{tagbox,taglayout}{%
4434   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
4435 \eqld@define@key{control}{tagbox*,taglayout*}{%
4436   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
4437 \eqld@define@key{control}{tagform}{%
4438   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
4439 \eqld@define@key{control}{tagform*}[#####1]{%
4440   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

**Equation Numbering.** Configure equation numbering schemes:

```

4441 \def\eqld@keycat{equations,setup}
4442 \eqld@define@key\eqld@keycat{numberline,number,num,numline,n}[all]{%
4443   \eqld@numbering@set{####1}}
4444 \eqld@define@key\eqld@keycat{nonumber,nn,*}[][%
4445   \let\eqld@numbering@active\eqld@false}
4446 \eqld@define@key\eqld@keycat{donumber,dn,!}[][%
4447   \let\eqld@numbering@active\eqld@true}

```

```

4448 \eqld@define@key\eqld@keycat{tagsleft,leqno}[]{\let\eqld@tagsleft\eqld@true}
4449 \eqld@define@key\eqld@keycat{tagsright,reqno}[]{\let\eqld@tagsleft\eqld@false}
4450 \eqld@define@key\eqld@keycat{tags,eqno}{%
4451   \eqld@decide@select{#3}{#2}{#1}{%
4452     {{right,r}}{\let\eqld@tagsleft\eqld@false}},%
4453     {{left,l}}{\let\eqld@tagsleft\eqld@true}}}}
4454 \eqld@define@key\eqld@keycat{bestlineauto}[true]{%
4455   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}

```

**TODO:** describe

```

4456 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
4457 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
4458 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
4459 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

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

```

4460 \def\eqld@keycat{equations,setup}
4461 \eqld@define@key\eqld@keycat{layout}{\eqld@decide@select{#3}{#2}{#1}{%
4462   {{center,c}}{\let\eqld@layoutleft\eqld@false}},%
4463   {{left,l}}{\let\eqld@layoutleft\eqld@true}}}}
4464 \eqld@define@key\eqld@keycat{center}[]{\let\eqld@layoutleft\eqld@false}
4465 \eqld@define@key\eqld@keycat{flushleft,left}[]{\let\eqld@layoutleft\eqld@true}
4466 \eqld@define@key\eqld@keycat{leftmargin}{\def\eqld@layoutleftmargin{#1}}
4467 \eqld@define@key\eqld@keycat{leftmargin*}{%
4468   \settowidth\dimen@{#1}\edef\eqld@layoutleftmargin{\the\dimen@}}
4469 \eqld@define@key\eqld@keycat{minleftmargin}{%
4470   \def\eqld@layoutleftmarginmin{#1}}
4471 \eqld@define@key\eqld@keycat{maxleftmargin}{%
4472   \eqld@decide@select{#3}{#2}{#1}{%
4473     {\eqld@decide@false{\def\eqld@layoutleftmarginmax{.5\maxdimen}}},%
4474     {\relax{\def\eqld@layoutleftmarginmax{#1}}}}}
4475 \def\eqld@keycat{equations,equationsbox}
4476 \eqld@define@key\eqld@keycat{margin}{%
4477   \def\eqld@display@marginleft{#1}\def\eqld@display@marginright{#1}}
4478 \eqld@define@key\eqld@keycat{marginleft}{\def\eqld@display@marginleft{#1}}
4479 \eqld@define@key\eqld@keycat{marginright}{\def\eqld@display@marginright{#1}}
4480 \eqld@define@key\eqld@keycat{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

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

```

4481 \def\eqld@keycat{equations,setup}
4482 \eqld@define@key\eqld@keycat{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
4483   {{max,full,4}}{\eqld@alignbadness@inf@bad}},%
4484   {{high,3}}{\eqld@alignbadness@54\relax}},%
4485   {{med,medium,2}}{\eqld@alignbadness@18\relax}},%
4486   {{low,1}}{\eqld@alignbadness@6\relax}},%
4487   {{0,\eqld@decide@false}}{\eqld@alignbadness@z@}}}}
4488 \eqld@define@key\eqld@keycat{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
4489   {{max,full,4}}{\eqld@tagbadness@inf@bad}},%
4490   {{high,3}}{\eqld@tagbadness@54\relax}},%
4491   {{med,medium,2}}{\eqld@tagbadness@18\relax}},%
4492   {{low,1}}{\eqld@tagbadness@6\relax}},%
4493   {{0,\eqld@decide@false}}{\eqld@tagbadness@z@}}}}

```

```

4494 \eqld@define@key\eqld@keycat{alignbadness}{\eqld@alignbadness@numexpr#1\relax}
4495 \eqld@define@key\eqld@keycat{tagbadness}{\eqld@tagbadness@numexpr#1\relax}
4496 \eqld@define@key\eqld@keycat{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
4497   {0,\eqld@decide@false}{\def\eqld@colsepmin@val{0pt}}},%
4498   {\relax{\def\eqld@colsepmin@val{#1}}}}}%
4499 \eqld@define@key\eqld@keycat{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
4500   {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
4501   {\relax{\def\eqld@colsepmax@val{#1}}}}}%
4502 \eqld@define@key\eqld@keycat{fulllength}[true]{%
4503   \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

4504 \eqld@define@key{equationsbox,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
4505   {0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
4506   {\relax{\def\eqld@box@colsep{#1}}}}}%
4507 \eqld@define@key{equations}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
4508   {0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
4509   {\relax{\def\eqld@box@colsep{#1}}}}}%
4510 \let\eqld@colsepmin@val\eqld@box@colsep
4511 \let\eqld@colsepmax@val\eqld@box@colsep

```

**Horizontal Shape.** Configure horizontal alignment schemes:

```

4512 \def\eqld@keycat{equations,equationsbox,setup}
4513 \eqld@define@key\eqld@keycat{shape}[default]{\eqld@shape@set{#1}}
4514 \eqld@define@key\eqld@keycat{padding,pad}[indent]{%
4515   \eqld@decide@select{#3}{#2}{#1}{%
4516     {\max}{\let\eqld@paddingleft@val\undefined}},%
4517     {\indent}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
4518     {0,\eqld@decide@false}{\def\eqld@paddingleft@val{0pt}}},%
4519     {\relax{\def\eqld@paddingleft@val{#1}}}}}%
4520 \let\eqld@paddingright@val\eqld@paddingleft@val
4521 \eqld@define@key\eqld@keycat{padleft}[indent]{%
4522   \eqld@decide@select{#3}{#2}{#1}{%
4523     {\max}{\let\eqld@paddingleft@val\undefined}},%
4524     {\indent}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
4525     {0,\eqld@decide@false}{\def\eqld@paddingleft@val{0pt}}},%
4526     {\relax{\def\eqld@paddingleft@val{#1}}}}}%
4527 \eqld@define@key\eqld@keycat{padright}[indent]{%
4528   \eqld@decide@select{#3}{#2}{#1}{%
4529     {\max}{\let\eqld@paddingright@val\undefined}},%
4530     {\indent}{\def\eqld@paddingright@val{\eqld@indent@val}}},%
4531     {0,\eqld@decide@false}{\def\eqld@paddingright@val{0pt}}},%
4532     {\relax{\def\eqld@paddingright@val{#1}}}}}%
4533 \eqld@define@key\eqld@keycat{indent}[2em]{%
4534   \def\eqld@indent@val{#1}}

```

**TODO:** describe

```

4535 \eqld@define@key{control}{align}[]{%
4536   \eqld@decide@select{#3}{#2}{#1}{%
4537     {\l,left}{\global\eqld@append\eqld@cell@container{\eqld@shape@pos@z}},%
4538     {\c,center}{\global\eqld@append\eqld@cell@container{\eqld@shape@pos@ne}},%
4539     {\r,right}{\global\eqld@append\eqld@cell@container{\eqld@shape@pos@tw}}}}}%
4540 \eqld@define@key{control}{shift,shifto}[]{%
4541   \eqld@decide@select{#3}{#2}{#1}{%
4542     {\*,indent}{\eqld@shape@alignamount@set{\eqld@indent@val}},%
4543     {\!,outdent}{\eqld@shape@alignamount@set{-\eqld@indent@val}},%
4544     {\relax{\eqld@shape@alignamount@set{#1}}}}}%
4545 \eqld@define@key{control}{shift*,shiftby}[]{\eqld@shape@alignamount@add{#1}}

```



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

```

4546 \def\eql@keycat{equations,equationsbox,setup}
4547 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
4548 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
4549 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
4550 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
4551 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
4552 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}{%
4553   {\ampeq,amprel,eqafter,beforerel}\eql@class@ampeq},%
4554   {\eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}}
```

**Punctuation.** Configure punctuation defaults:

```

4555 \def\eql@keycat{equations,equationsbox,setup}
4556 \eql@define@key\eql@keycat{punctsep}[\,]{\def\eql@punct@sep{#1}}
4557 \eql@define@key\eql@keycat{punct}[.]{\def\eql@punct@main{#1}}
4558 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
4559 \eql@define@key\eql@keycat{punctline}[,]{\def\eql@punct@line{#1}}
4560 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
4561 \eql@define@key\eql@keycat{punctcol}[,]{\def\eql@punct@col{#1}}
4562 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}

4563 \eql@define@key{control}{punctsep}[\,]{\def\eql@punct@sep{#1}}
4564 \eql@define@key{control}{punct}[.]{\def\eql@punct@block{#1}%
4565   \def\eql@punct@line{#1}\def\eql@punct@col{#1}}
4566 \eql@define@key{control}{punct*}[]{\let\eql@punct@block\relax}
4567 \eql@define@key{control}{punctapply}[]{\eql@punct@apply@block}
```

**Frames.** **TODO:** describe

```

4568 \eql@define@key{equationsbox}{frame}[\fbox]{%
4569   \def\eql@box@frame{#1}%
4570   \ifx\eql@box@frame\@empty\let\eql@box@frame\@firstofone\fi}
4571 \eql@define@key{equationsbox}{wrap}[]{\eql@box@wrap#1}
```

**TODO:** describe

```

4572 \eql@define@key{control}{framecell}[\fbox]{%
4573   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}}
4574 \eql@define@key{control}{frametag}[\fbox]{%
4575   \global\eql@append\eql@tags@container{\def\eql@tags@frame@cmd{#1}}}
```

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

```

4576 \eql@define@key{equations,equationsbox}{alt}{}
```

**Global Switches.** Set global switches:

```

4577 \let\eql@multi@linesfallback\eql@false
4578 \let\eql@scan@par\eql@false
4579 \let\eql@single@crerror\eql@false
4580 \let\eql@ampproof@active\eql@false

4581 \eql@define@key{equations,setup}{linesfallback}[true]{%
4582   \eql@decide@select{#3}{#2}{#1}{%
4583     {\eql@decide@false{\let\eql@multi@linesfallback\eql@false}},%
```

```

4584     {\reuse,lean}{\let\eq@multi@linesfallback\z@}},%
4585     {\measure,full,\eq@decide@true}{\let\eq@multi@linesfallback\eq@true}}}}
4586 \eq@define@key{setup}{ampproof}[true]{%
4587   \eq@decide@bool{#3}{#2}{#1}\eq@ampproof@active}
4588 \eq@define@key{setup}{crrerror}[true]{%
4589   \eq@decide@bool{#3}{#2}{#1}\eq@single@crrerror}
4590 \eq@define@key{setup}{modifierwarning}[true]{%
4591   \eq@decide@select{#3}{#2}{#1}{%
4592     {\eq@decide@false}{\let\eq@parseopt@warn\@empty}},%
4593     {\eq@decide@true}{\let\eq@parseopt@warn\eq@warn@parseopt}},%
4594     {\verbose,+}{\let\eq@parseopt@warn\eq@warn@parseopt@verbose}}}}
4595 \let\eq@parseopt@warn\eq@warn@parseopt
4596 \eq@define@key{equations,setup}{rescan}[true]{%
4597   \eq@decide@if{#3}{#2}{#1}%
4598   {\let\eq@scan@body\eq@scan@body@rescan}%
4599   {\let\eq@scan@body\eq@scan@body@dump}}
4600 \eq@define@key{equations,equationsbox,setup}{scanpar}[true]{%
4601   \eq@decide@bool{#3}{#2}{#1}\eq@scan@par}
4602 \eq@define@key{setup}{defaults}{%
4603   \eq@decide@select{#3}{#2}{#1}{%
4604     {\classic}{\eq@defaults@classic}},%
4605     {\eqnlines}{\eq@defaults@eqnlines}}}}

```

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

```

4606 \let\eq@provide@opt@env\tw@
4607 \let\eq@provide@opt@amsmathends\eq@true
4608 \let\eq@provide@opt@backup\eq@false
4609 \let\eq@provide@opt@ang\eq@true
4610 \let\eq@provide@opt@eqref\eq@true

4611 \eq@define@key{setup}{amsmathends}[true]{%
4612   \eq@error@packageoption{#2}%
4613   \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@amsmathends}
4614 \eq@define@key{setup}{backup}[true]{%
4615   \eq@error@packageoption{#2}%
4616   \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@backup}
4617 \eq@define@key{setup}{env}[equation]{%
4618   \eq@error@packageoption{#2}%
4619   \eq@decide@select{#3}{#2}{#1}{%
4620     {\none,\eq@decide@false}{\let\eq@provide@opt@env\z@}},%
4621     {\equation,latex}{\let\eq@provide@opt@env@cne}},%
4622     {\amsmath,all,\eq@decide@true}{\let\eq@provide@opt@env\tw@}}}}
4623 \eq@define@key{setup}{ang}[true]{%
4624   \eq@error@packageoption{#2}%
4625   \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@ang}
4626 \eq@define@key{setup}{eqref}[true]{%
4627   \eq@error@packageoption{#2}%
4628   \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@eqref}

```

**Shortcut Options.** **TODO:** describe

```

4629 \def\eq@parseopt@nonumber#1{\eqnadopt{nonumber}\eq@parseopt@peek}
4630 \def\eq@parseopt@donumber#1{\eqnadopt{donumber}\eq@parseopt@peek}
4631 \def\eq@parseopt@single#1{\eqnadopt{single}\eq@parseopt@peek}
4632 \def\eq@parseopt@lines#1{\eqnadopt{lines}\eq@parseopt@peek}
4633 \def\eq@parseopt@columns#1{\eqnadopt{columns}\eq@parseopt@peek}
4634 \def\eq@parseopt@transpose#1{\eqnadopt{columns,transpose}\eq@parseopt@peek}

```

```

4635 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
4636 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
4637 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
4638 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
4639 \def\eql@parseopt@punctoff#1{\eqnaddopt{punct={}}\eql@parseopt@peek}

```

### P.3 Parameter Presets

The package offers two parameter presets which lead to somewhat different layout. Instead of setting the internal parameters directly, we expose them as public settings so that they are easier to read and such that individual settings can be used to compose own layouts.

`eql@defaults@classic` The preset `classic` aims to reproduce the  $\mathrm{T}_\mathrm{E}\mathrm{X}$ ,  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}\mathrm{X}$  and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

4640 \def\eql@defaults@classic{%
4641   \eqnlineset{numberline=all}%
4642   \eqnlineset{mintagsep={.5\fontdimen6\textfont2\relax}}%
4643   \eqnlineset{maxcolsep=off}%
4644   \eqnlineset{spread={\jot}}%
4645   \eqnlineset{tagmargin}%
4646   \eqnlineset{tagmarginratio=1}%
4647   \eqnlineset{tagmarginthreshold=0.5}%
4648   \eqnlineset{leftmargin={\leftmargini}}%
4649   \eqnlineset{padding=max}%
4650   \eqnlineset{bestlineauto=off}%
4651   \eqnlineset{displayheight=off}%
4652   \eqnlineset{displaydepth=off}%
4653   \eqnlineset{shortmode=belowsingle}%
4654   \eqnlineset{abovecontmode=short}%
4655   \eqnlineset{belowcontmode=short}%
4656   \eqnlineset{aboveparmode=long}%
4657   \eqnlineset{belowparmode=long}%
4658   \eqnlineset{abovetopmode=long}%
4659   \eqnlineset{belowtopmode=long}%
4660   \eqnlineset{abovelongskip={\abovedisplayskip}}%
4661   \eqnlineset{belowlongskip={\belowdisplayskip}}%
4662   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
4663   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
4664   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
4665   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
4666   \eqnlineset{abovecontskip=0pt}%
4667   \eqnlineset{belowcontskip=0pt}%
4668   \eqnlineset{aboveparskip=0pt}%
4669   \eqnlineset{belowparskip=0pt}%
4670   \eqnlineset{abovetopskip=0pt}%
4671   \eqnlineset{belowtopskip=0pt}%
4672   \eqnlineset{abovetagskip=0pt}%
4673   \eqnlineset{belowtagskip=0pt}%
4674   \eqnlineset{abovemedtagskip=0pt}%
4675   \eqnlineset{belowmedtagskip=0pt}%
4676   \eqnlineset{abovepartagskip=0pt}%
4677   \eqnlineset{belowpartagskip=0pt}%
4678   \eqnlineset{crerror=false}%
4679   \eqnlineset{linesfallback=false}%
4680 }

```

values based on 10pt vs 12pt

`eqn@defaults@eqnlines` The (default) preset `eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the `classic` spacing for a 12pt computer modern font such that 10pt fonts will lead to slightly reduced spacing. Apart from that, the `eqnlines` setting makes some deliberate layout choices that deviate significantly from `classic` (maximum column separation, no shortening below equations):

```

4681 \def\eqn@defaults@eqnlines{%
4682   \eqnlineset{numberline=all}%
4683   \eqnlineset{mintagsep=.5em}%
4684   \eqnlineset{maxcolsep=2em}%
4685   \eqnlineset{spread={0.2\normalbaselineskip}}%
4686   \eqnlineset{tagmargin}%
4687   \eqnlineset{tagmarginratio=.334}%
4688   \eqnlineset{tagmarginthreshold=0.5}%
4689   \eqnlineset{leftmargin={\leftmargini}}%
4690   \eqnlineset{padding=0pt}%
4691   \eqnlineset{bestlineauto}%
4692   \eqnlineset{displayheight=strut}%
4693   \eqnlineset{displaydepth=strut}%
4694   \eqnlineset{shortmode=above}%
4695   \eqnlineset{abovecontmode=noskip}%
4696   \eqnlineset{belowcontmode=long}%
4697   \eqnlineset{aboveparmode=long}%
4698   \eqnlineset{belowparmode=long}%
4699   \eqnlineset{abovetopmode=noskip}%
4700   \eqnlineset{belowtopmode=long}%
4701   \eqnlineset{longskip={0.75\normalbaselineskip
4702     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
4703   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
4704     plus 0.25\normalbaselineskip}}%
4705   \eqnlineset{belowshortskip={0.0\normalbaselineskip
4706     plus 0.25\normalbaselineskip}}%
4707   \eqnlineset{medskip={0.4\normalbaselineskip
4708     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
4709   \eqnlineset{abovecontskip=0pt}%
4710   \eqnlineset{belowcontskip=0pt}%
4711   \eqnlineset{aboveparskip=0pt}%
4712   \eqnlineset{belowparskip=0pt}%
4713   \eqnlineset{abovetopskip=0pt}%
4714   \eqnlineset{belowtopskip=0pt}%
4715   \eqnlineset{abovetagskip=0pt}%
4716   \eqnlineset{belowtagskip=0pt}%
4717   \eqnlineset{abovemedtagskip=0pt}%
4718   \eqnlineset{belowmedtagskip=0pt}%
4719   \eqnlineset{abovepartagskip=0pt}%
4720   \eqnlineset{belowpartagskip=0pt}%
4721   \eqnlineset{crerror=true}%
4722   \eqnlineset{linesfallback=true}%
4723 }
```

## P.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of `LATEX` and `amsmath` carefully.

### Tools.

`\eql@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```

\eq@provide@moveenv 4724 \def\eql@provide@movecmd#1#2{%
eq@provide@movestar 4725   \eq@letcs{#1\expandafter}\csname #2\endcsname
@provide@undefinecmd 4726 }
@provide@undefineenv 4727 \def\eql@provide@moveenv#1#2{%
                     4728   \eq@provide@movecmd{#1}{#2}%
                     4729   \ifcsname end#2\endcsname
                     4730     \eq@provide@movecmd{end#1}{end#2}%
                     4731   \fi
                     4732 }
                     4733 \def\eql@provide@movestar#1#2{%
                     4734   \eq@provide@moveenv{#1}{#2}%
                     4735   \ifcsname #2*\endcsname
                     4736     \eq@provide@moveenv{#1*}{#2*}%
                     4737   \fi
                     4738 }
                     4739 \def\eql@provide@undefinecmd#1{%
                     4740   \eq@letcs{#1}\undefined
                     4741 }
                     4742 \def\eql@provide@undefineenv#1{%
                     4743   \eq@provide@undefinecmd{#1}%
                     4744   \eq@provide@undefinecmd{end#1}%
                     4745 }

```

**Fix Endings for amsmath Environments.** The amsmath derived environments forward their ending routines directly to the ending routines for the main environments `gather`, `multline`, `align`, `aligned`. This causes a problem when the main environments are replaced but the derived ones are still used. We fix the potential problem by copying the ending routines of the main environments to the ending routines of the derived environments.

`\eql@amsmath@endfix` Check whether the original forwarding of an ending routine is still in place (other packages or future updates to amsmath might change the behaviour). If so, copy the ending routine into place:

```

4746 \def\eql@amsmath@endfix#1#2{%
4747   \long\edef\@tempa{\expandafter\noexpand\csname end#2\endcsname}%
4748   \expandafter\ifx\csname end#1\endcsname\@tempa
4749     \eq@provide@movecmd{end#1}{end#2}%
4750   \fi
4751 }

```

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

```

4752 \def\eql@amsmath@fixends{%
4753   \eq@amsmath@after{%
4754     \eq@amsmath@endfix{gather*}{gather}%
4755     \eq@amsmath@endfix{multline*}{multline}%
4756     \eq@amsmath@endfix{align*}{align}%
4757     \eq@amsmath@endfix{flalign*}{align}%
4758     \eq@amsmath@endfix{alignat*}{align}%
4759     \eq@amsmath@endfix{alignat*}{align}%
4760     \eq@amsmath@endfix{xalignat*}{align}%
4761     \eq@amsmath@endfix{xalignat*}{align}%
4762     \eq@amsmath@endfix{xxalignat*}{align}%
4763     \eq@amsmath@endfix{gathered}{aligned}%
4764   }

```

```

4765 \eql@amsmath@endfix{alignedat}{aligned}%
4766 }
4767 }

```

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

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

```

4768 \def\eql@provide@backup@amsmath#1{%
4769 \eql@amsmath@after{%
4770 \eql@provide@moveenv{ams#1}{#1}%
4771 }%
4772 }

```

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

```

4773 \def\eql@provide@backup@eqref{%
4774 \eql@amsmath@after{%
4775 \eql@provide@movecmd{amseqref}{eqref}%
4776 }%
4777 }

```

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

```

4778 \def\eql@provide@backup@multlined{%
4779 \AddToHook{package/mathtools/after}{%
4780 \eql@provide@moveenv{amsmultlined}{multlined}%
4781 }%
4782 }

```

`vide@backup@equation` The  $\text{\LaTeX}$  environment `equation` is overwritten by several packages to implement their adjustments. Here we cater for adjustments through `amsmath`, `hyperref` and the PDF tagging mechanism. Copy `equation` and `equation*` whenever `amsmath` is loaded. Whenever `hyperref` is loaded, and `amsmath` is not yet present, backup the original  $\text{\LaTeX}$  and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original  $\text{\LaTeX}$  `equation`. The PDF tagging mechanism registers `equation` upon `\begin{document}`. We thus need to register all copies of `equation` on our own, so that they can be used with their new names:

```

4783 \def\eql@provide@backup@equation{%
4784 \eql@amsmath@after{%
4785 \eql@provide@moveenv{amsequation}{equation}%
4786 \eql@tagging@register@env{amsequation}%
4787 \eql@provide@moveenv{amsequation*}{equation*}%
4788 \eql@tagging@register@env{amsequation*}%
4789 }%
4790 \AddToHook{package/hyperref/after}{%
4791 \@ifpackageloaded{amsmath}{}%
4792 \let\latexequation\H@equation
4793 \let\endlatexequation\H@endequation
4794 \eql@tagging@register@env{latexequation}%
4795 \eql@provide@moveenv{hyperrefequation}{equation}%
4796 \eql@tagging@register@env{hyperrefequation}%
4797 }%
4798 }%

```

```

4799 \@ifpackageloaded{amsmath}{\@ifpackageloaded{hyperref}}{%
4800   \eql@provide@moveenv{latexequation}{equation}%
4801   \eql@tagging@register@env{latexequation}%
4802 }}%
4803 }

```

e@backup@displaymath **TODO:** describe

```

4804 \def\eql@provide@backup@displaymath{%
4805   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
4806 }

```

@backup@subequations The amsmath subequations environment is adjusted by hyperref through an environment hook, but this hook gets applied only later at `\begin{document}`. Hence, we need to supply the hook routine to the new routine ourselves:

```

4807 \def\eql@provide@backup@subequations{%
4808   \eql@amsmath@after{%
4809     \eql@provide@moveenv{amssubequations}{subequations}%
4810   }%
4811   \AddToHook{package/hyperref/after}{%
4812     \AddToHook{cmd/amssubequations/before}%
4813     {%
4814       \stepcounter{equation}%
4815       \protected@edef\theHparentequation{\theHequation}%
4816       \addtocounter{equation}{-1}%
4817     }%
4818     \AddToHook{cmd/amssubequations/after}%
4819     {%
4820       \def\theHequation{\theHparentequation\alph{equation}}%
4821       \ignorespaces
4822     }%
4823   }%
4824 }

```

\eql@provide@backup Backup all amsmath environments:

```

4825 \def\eql@provide@backup{%
4826   \eql@provide@backup@eqref
4827   \eql@provide@backup@equation
4828   \eql@provide@backup@displaymath
4829   \eql@provide@backup@amsmath@gather}%
4830   \eql@provide@backup@amsmath@gather*}%
4831   \eql@provide@backup@amsmath{multline}%
4832   \eql@provide@backup@amsmath{multline*}%
4833   \eql@provide@backup@amsmath{align}%
4834   \eql@provide@backup@amsmath{align*}%
4835   \eql@provide@backup@amsmath{flalign}%
4836   \eql@provide@backup@amsmath{flalign*}%
4837   \eql@provide@backup@amsmath{alignat}%
4838   \eql@provide@backup@amsmath{alignat*}%
4839   \eql@provide@backup@amsmath{xalignat}%
4840   \eql@provide@backup@amsmath{xalignat*}%
4841   \eql@provide@backup@amsmath{xxalignat}%
4842   \eql@provide@backup@amsmath{aligned}%
4843   \eql@provide@backup@amsmath{aligned*}%
4844   \eql@provide@backup@amsmath{alignedat}%
4845   \eql@provide@backup@amsmath{alignedat*}%
4846   \eql@provide@backup@amsmath{gathered}%

```

```

4847 \eql@provide@backup@amsmath{gathered*}%
4848 \eql@provide@backup@multlined
4849 \eql@provide@backup@subequations
4850 }

```

**Replacement amsmath Environments.** **TODO:** describe

```

4851 \def\eql@alignat@gobblecol#1{%
4852 \eql@ifnextchar@loose\bgroup{\@firstoftwo{#1}}{#1}}

```

**eql@gathered** (*env.*) Define replacement versions for boxed environments **gathered**, **multlined** and **aligned**  
**eql@multlined** (*env.*) which forward to **equationsbox** with specific presets:

```

eql@aligned (env.)
4853 \newenvironment{eql@gathered}
4854 {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
4855 \newenvironment{eql@multlined}
4856 {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
4857 \newenvironment{eql@aligned}
4858 {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
4859 \newenvironment{eql@alignedat}
4860 {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
4861 {\endequationsbox}

```

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

```

eql@align (env.)
4862 \newenvironment{eql@equation}
4863 {\eqnaddopt{equation}\equations}{\endequations}
4864 \newenvironment{eql@displaymath}
4865 {\eqnaddopt{equation,nonumber}\equations}{\endequations}
4866 \newenvironment{eql@gather}
4867 {\eqnaddopt{lines}\equations}{\endequations}
4868 \newenvironment{eql@multline}
4869 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
4870 {\endequations}
4871 \newenvironment{eql@align}
4872 {\eqnaddopt{columns}\equations}{\endequations}
4873 \newenvironment{eql@flalign}
4874 {\eqnaddopt{fulllength}\eql@align}{\endequations}
4875 \newenvironment{eql@alignat}
4876 {\eqnaddopt{colsep=off}\eql@xalignat}{\endequations}
4877 \newenvironment{eql@xalignat}
4878 {\eql@alignat@gobblecol\eql@align}{\endequations}
4879 \newenvironment{eql@xxalignat}
4880 {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
4881 \newenvironment{eql@equation*}
4882 {\eqnaddopt{nonumber}\eql@equation}{\endequations}
4883 \newenvironment{eql@gather*}
4884 {\eqnaddopt{nonumber}\eql@gather}{\endequations}
4885 \newenvironment{eql@multline*}
4886 {\eqnaddopt{nonumber}\eql@multline}{\endequations}
4887 \newenvironment{eql@align*}
4888 {\eqnaddopt{nonumber}\eql@align}{\endequations}
4889 \newenvironment{eql@flalign*}
4890 {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
4891 \newenvironment{eql@alignat*}
4892 {\eqnaddopt{nonumber}\eql@alignat}{\endequations}

```



```

4893 \newenvironment{eql@xalignat*}
4894 {\eqnaddopt{nonumber}\eql@xalignat}{\endequations}

```

**Install Additional Environments.** The additional environments need to be installed at their intended names which can be adjusted by the user.

**eql@provide@onlyonce** Process arguments for providing a specific environment. #1 describes the environment using the amsmath name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the amsmath environment in place making sure that the replacement is robust against loading amsmath before or after. If #2 equals '\*', just overwrite the amsmath environment in place immediately (e.g. within a block in the document body):

```

4895 \def\eql@provide@onlyonce#1#2{%
4896   \def\eql@tmp{#2}%
4897   \def\@tempa{#1}%
4898   \ifx\eql@tmp\@tempa
4899     \let\eql@tmp\@empty
4900   \fi
4901   \ifx\eql@tmp\@empty
4902     \let\eql@tmp\@undefined
4903     \ifx\@nodocument\relax
4904       \def\eql@tmp{#1}%
4905     \fi
4906     \ifcsname eql@provided@#1\endcsname
4907       \def\eql@tmp{#1}%
4908     \fi
4909     \eql@letcs{eql@provided@#1}\eql@true
4910   \else
4911     \def\@tempa{*}%
4912     \ifx\eql@tmp\@tempa
4913       \def\eql@tmp{#1}%
4914     \fi
4915   \fi
4916 }

```

**\eql@provide@eqref** Provide \eqref as the macro #1. We have to check whether #1 is empty or equals \eqref or takes the value '\*'. If not, we should strip the backslash for further processing. Copy the macro into place, and copy again when amsmath or mathtools are loaded. Remove definition before amsmath is loaded in the future to avoid a potential error:

```

4917 \def\eql@provide@eqref#1{%
4918   \def\eql@tmp{#1}%
4919   \def\@tempa{\eqref}%
4920   \ifx\eql@tmp\@tempa
4921     \let\eql@tmp\@empty
4922   \fi
4923   \ifx\eql@tmp\@empty
4924     \eql@provide@onlyonce{eqref}{}%
4925   \else
4926     \def\@tempa{*}%
4927     \ifx\eql@tmp\@tempa
4928       \def\eql@tmp{eqref}%
4929     \else
4930       \edef\eql@tmp{\expandafter\@gobble\string#1}%
4931     \fi
4932   \fi
4933   \ifdefined\eql@tmp

```

```

4934 \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
4935 \else
4936 \eql@amsmath@after{%
4937 \eql@provide@movecmd{eqref}{eql@eqref}%
4938 }%
4939 \AddToHook{package/mathtools/after}{%
4940 \eql@provide@movecmd{eqref}{eql@eqref}%
4941 }%
4942 \eql@provide@movecmd{eqref}{eql@eqref}%
4943 \eql@amsmath@undefine\eqref
4944 \fi
4945 }

```

`\eql@provide@amsmath` Provide one of the `amsmath` environments and its star variant. Copy into place, and copy again when `amsmath` or `mathtools` are loaded. Remove definition before `amsmath` is loaded in the future to avoid an error:

```

4946 \def\eql@provide@amsmath#1#2{%
4947 \eql@provide@onlyonce{#1}{#2}%
4948 \ifdefined\eql@tmp
4949 \expandafter\eql@provide@movestart\expandafter{\eql@tmp}{eql@#1}%
4950 \else
4951 \eql@amsmath@after{%
4952 \eql@provide@movestart{#1}{eql@#1}%
4953 }%
4954 \AddToHook{package/mathtools/after}{%
4955 \eql@provide@movestart{#1}{eql@#1}%
4956 }%
4957 \eql@provide@movestart{#1}{eql@#1}%
4958 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
4959 \ifcsname eql@#1*\endcsname
4960 \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
4961 \fi
4962 \fi
4963 }

```

`\eql@provide@multlined` Provide `mathtools` environment `multlined`. Copy into place, and copy again when `amsmath` or `mathtools` are loaded. Remove definition before `mathtools` is loaded in the future to avoid an error:

```

4964 \def\eql@provide@multlined#1{%
4965 \eql@provide@onlyonce{multlined}{#1}%
4966 \ifdefined\eql@tmp
4967 \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
4968 \else
4969 \AddToHook{package/mathtools/after}{%
4970 \eql@provide@moveenv{multlined}{eql@multlined}%
4971 }%
4972 \eql@provide@moveenv{multlined}{eql@multlined}%
4973 \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
4974 \eql@provide@undefineenv{multlined}}}%
4975 \fi
4976 }

```

`\eql@provide@equation` Provide the environment `equation` and its star variant. Copy into place, and copy again when `amsmath` or `hyperref` are loaded. Remove definition of `equation*` before `amsmath` is loaded in the future to avoid an error. When PDF tagging is active, the environment is modified at `\begin{document}` in an undesirable fashion, so copy the definition again:

```

4977 \def\eql@provide@equation#1{%
4978   \eql@provide@onlyonce{equation}{#1}%
4979   \ifdefined\eql@tmp
4980     \expandafter\eql@provide@movestart\expandafter{\eql@tmp}{eql@equation}%
4981   \else
4982     \eql@amsmath@after{%
4983       \eql@provide@movestart{equation}{eql@equation}%
4984     }%
4985     \AddToHook{package/hyperref/after}{%
4986       \@ifpackageloaded{amsmath}{}{%
4987         \eql@provide@moveenv{equation}{eql@equation}%
4988       }%
4989     }%
4990     \eql@provide@movestart{equation}{eql@equation}%
4991     \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
4992     \ifdefined\eql@tagging@on
4993       \AddToHook{begindocument/end}{%
4994         \eql@provide@movestart{equation}{eql@equation}%
4995       }%
4996     \fi
4997   \fi
4998 }

```

`@provide@displaymath` **TODO:** describe

```

4999 \def\eql@provide@displaymath#1{%
5000   \eql@provide@onlyonce{displaymath}{#1}%
5001   \ifdefined\eql@tmp
5002     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@displaymath}%
5003   \else
5004     \eql@provide@moveenv{displaymath}{eql@displaymath}%
5005     \ifdefined\eql@tagging@on
5006       \AddToHook{begindocument/end}{%
5007         \eql@provide@moveenv{displaymath}{eql@displaymath}%
5008       }%
5009     \fi
5010   \fi
5011 }

```

`provide@subequations` Provide the `amsmath` environment `subequations`. Copy into place, and copy again when `amsmath` is loaded. `hyperref` adds a hook to the command which messes up the parsing of optional arguments (even if the hook is emptied). The hook placement happens at `\begin{document}`, so we copy the environment again afterwards. We also remove the hook (after adding an empty hook to avoid errors). Remove definition before `amsmath` is loaded in the future to avoid an error:

```

5012 \def\eql@provide@subequations#1{%
5013   \eql@provide@onlyonce{subequations}{#1}%
5014   \ifdefined\eql@tmp
5015     \expandafter\eql@provide@moveenv
5016     \expandafter{\eql@tmp}{eql@subequations}%
5017   \else
5018     \eql@amsmath@after{%
5019       \eql@provide@moveenv{subequations}{eql@subequations}%
5020     }%
5021     \AddToHook{package/hyperref/after}{%
5022       \AddToHook{cmd/subequations/before}[hyperref]{}%
5023       \AddToHook{cmd/subequations/after}[hyperref]{}%

```

```

5024 \RemoveFromHook{cmd/subequations/before}[hyperref]%
5025 \RemoveFromHook{cmd/subequations/after}[hyperref]%
5026 \AddToHook{begindocument/end}{%
5027 \eql@provide@moveenv{subequations}{eql@subequations}%
5028 }%
5029 }%
5030 \eql@provide@moveenv{subequations}{eql@subequations}%
5031 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
5032 \fi
5033 }

```

`\eql@provide@sqr` Provide the symbolic environment `\[...]`. Copy into place, and copy again when `amsmath` is loaded. If PDF tagging is active, some undesired modifications happen at `\begin{document}`, so copy again afterwards:

```

5034 \def\eql@provide@sqr{%
5035 \let\[\eql@sqr@open
5036 \let\]\eql@sqr@close
5037 \eql@amsmath@after{%
5038 \let\[\eql@sqr@open
5039 \let\]\eql@sqr@close
5040 }%
5041 \ifdefined\eql@tagging@on
5042 \AddToHook{begindocument/end}{%
5043 \let\[\eql@sqr@open
5044 \let\]\eql@sqr@close
5045 }%
5046 \fi
5047 }

```

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

```

5048 \def\eql@provide@ang{%
5049 \let\<\eql@ang@open
5050 \let\>\eql@ang@close
5051 }

```

## Interface.

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

```

5052 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5053 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5054 \eql@define@key{provide}{gather}[]{\eql@provide@amsmath{gather}{#1}}
5055 \eql@define@key{provide}{multline}[]{\eql@provide@amsmath{multline}{#1}}
5056 \eql@define@key{provide}{align}[]{\eql@provide@amsmath{align}{#1}}
5057 \eql@define@key{provide}{flalign}[]{\eql@provide@amsmath{flalign}{#1}}
5058 \eql@define@key{provide}{alignat}[]{\eql@provide@amsmath{alignat}{#1}}
5059 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsmath{xalignat}{#1}}
5060 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsmath{xxalignat}{#1}}
5061 \eql@define@key{provide}{aligned}[]{\eql@provide@amsmath{aligned}{#1}}
5062 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsmath{alignedat}{#1}}
5063 \eql@define@key{provide}{gathered}[]{\eql@provide@amsmath{gathered}{#1}}
5064 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5065 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5066 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}

```

```

5067 \eqld@define@key{provide}{ang}[]{\eqld@provide@ang}
5068 \eqld@define@key{provide}{eqref}[]{\eqld@provide@eqref{#1}}
5069 \eqld@define@key{provide}{tagform}[]{%
5070   \def\tagform@##1{\maketag@@@{\eqld@tags@tagform{#1}}}}
5071 \eqld@define@key{provide}{maketag}[]{%
5072   \def\maketag@@@##1{\eqld@tags@taglayout{##1}}}

```

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

```

5073 \newcommand{\eqnlinesprovide}[1]{%
5074   (dev)\eqld@dev@start\eqnlinesprovide
5075   \eqld@setkeys{provide}{#1}%
5076   \ignorespaces
5077 }

```

## P.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
5078 \let\eqld@error@packageoption\@gobble
```

Declare math layout options `leqno` and `fleqn` for common L<sup>A</sup>T<sub>E</sub>X classes:

```

5079 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
5080 \DeclareOption{fleqn}{\eqnlineset{left}}

```

Pass undeclared options on to keyval processing:

```
5081 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

5082 \eqld@defaults@eqnlines
5083 \eqld@mode@columns
5084 \eqld@mode@aligned

```

Process package options:

```
5085 \ProcessOptions
```

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

```

5086 \def\eqld@error@packageoption#1{%
5087   \eqld@error{may only use '1' as a package option}%
5088 }

```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the T<sub>E</sub>X conditional parsing mechanism:

```

5089 \ifdefined\tagsleft@true\else
5090   \expandafter\newif\csname iftagsleft@\endcsname
5091 \fi
5092 \ifdefined\@fleqntrue\else
5093   \expandafter\newif\csname if@fleqn\endcsname
5094 \fi

```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`: **TODO**: this overrides user settings!

```
5095 \ifnum\eqld@provide@opt@env=\tw@
```

```

5096 \eql@amsmath@after{%
5097 \iftagsleft@
5098 \eqnlineset{tagsleft}
5099 \else
5100 \eqnlineset{tagsright}
5101 \fi
5102 \if@fleqn
5103 \eqnlineset{left}
5104 \else
5105 \eqnlineset{center}
5106 \fi
5107 }
5108 \fi

```

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

```
5109 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

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

```
5110 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

Provide native  $\LaTeX$  environment `equation` and symbolic shortcut `\[...\]` if desired:

```

5111 \ifnum\eql@provide@opt@env>\z@
5112 \eqnlinesprovide{equation,sqr,displaymath}
5113 \fi

```

Provide `amsmath` equation environments if desired:

```

5114 \ifnum\eql@provide@opt@env=\tw@
5115 \eqnlinesprovide{%
5116 multline,gather,align,flalign,alignat,xalignat,xxalignat,%
5117 multlined,gathered,aligned,alignedat,%
5118 subequations}
5119 \fi

```

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

```
5120 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
5121 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```