

The `eqnlines` Package

Source Code Documentation

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland
`nbeisert@itp.phys.ethz.ch`

v1.0: 2026-02-15

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

Abstract

`eqnlines` is a \LaTeX 2 ϵ package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of \LaTeX and the `amsmath` package with many options for convenient adjustment of the intended layout. In particular, the package adds flexible schemes for numbering, horizontal alignment and semi-automatic punctuation, and it improves upon the horizontal and vertical spacing options. The extensions can be used and adjusted through optional arguments and modifiers to the equation environments as well as global settings.

Contents

| | | |
|----------|--|-----------|
| 1 | Implementation | 3 |
| 2 | General Support | 3 |
| 2.1 | Debugging Messages | 4 |
| 2.2 | Supporting Definitions | 4 |
| 2.3 | Dollardollar Abstraction | 4 |
| 2.4 | Look-Ahead in Alignment | 5 |
| 2.5 | Error Messages | 7 |
| 2.6 | <code>amsmath</code> Integration | 8 |
| 2.7 | PDF Tagging Support | 8 |
| 2.8 | Key-Value Processing | 10 |
| 2.9 | Scanning the Equation Body | 12 |
| 3 | Parameters and Registers | 16 |
| 3.1 | Parameters | 17 |
| 3.2 | Registers | 20 |
| 3.3 | Hooks | 22 |
| 4 | Features | 22 |
| 4.1 | Punctuation | 22 |

| | | |
|-----------|---|-----------|
| 4.2 | Math Classes at Alignment | 25 |
| 4.3 | Framed Cells | 27 |
| 4.4 | Single-Line Composition | 27 |
| 4.5 | Alternative Content Description | 29 |
| 5 | Equation Numbering | 30 |
| 5.1 | Supporting Definitions | 30 |
| 5.2 | Schemes | 31 |
| 5.3 | Interface | 34 |
| 5.4 | Integration | 39 |
| 5.5 | Positioning | 42 |
| 5.6 | Component Display | 46 |
| 5.7 | Tag Composition | 49 |
| 6 | Subequation Numbering | 51 |
| 6.1 | Definitions | 51 |
| 6.2 | Environment | 52 |
| 6.3 | Subequation Scheme | 54 |
| 7 | Display Equations Support | 54 |
| 7.1 | Display Breaks | 55 |
| 7.2 | Explicit Vertical Space | 56 |
| 7.3 | Default Vertical Spacing | 57 |
| 7.4 | Entry and Exit | 58 |
| 7.5 | Stack | 64 |
| 8 | Multi-Line Support | 65 |
| 8.1 | Measure Support | 65 |
| 8.2 | Line Breaks | 67 |
| 8.3 | Intertext | 69 |
| 8.4 | Line Marks | 71 |
| 9 | Column Placement | 73 |
| 9.1 | Supporting Definitions | 73 |
| 9.2 | Shape Schemes | 73 |
| 9.3 | Width Data | 76 |
| 9.4 | Best Line Selection | 80 |
| 9.5 | Tag Margin | 81 |
| 9.6 | Single Column | 82 |
| 9.7 | Multiple Columns | 83 |
| 10 | Single Column Arrangement | 88 |
| 10.1 | Supporting Definitions | 88 |
| 10.2 | Arrangement Methods | 89 |
| 10.3 | Central Alignment | 90 |
| 10.4 | Left Alignment | 92 |
| 10.5 | Right Alignment | 93 |
| 11 | Equations Box Environment | 94 |
| 11.1 | Line Breaks | 94 |
| 11.2 | Column Breaks | 96 |
| 11.3 | Lines Mode | 98 |
| 11.4 | Columns Mode | 99 |
| 11.5 | Cases Mode | 101 |
| 11.6 | Main | 102 |

| | |
|--|------------|
| 11.7 Options Processing | 104 |
| 11.8 Environment | 106 |
| 12 Single-Line Equation | 107 |
| 12.1 Native Mode | 107 |
| 12.2 Print | 108 |
| 13 Multi-Line with Single Column | 109 |
| 13.1 Measure | 109 |
| 13.2 Column Placement | 111 |
| 13.3 Print | 111 |
| 14 Multi-Line with Multiple Columns | 113 |
| 14.1 Support | 113 |
| 14.2 Column Breaks | 113 |
| 14.3 Transpose | 115 |
| 14.4 Measure | 117 |
| 14.5 Columns Placement | 119 |
| 14.6 Print | 120 |
| 15 Interface | 123 |
| 15.1 Options Processing | 123 |
| 15.2 Single-Line Main | 125 |
| 15.3 Multi-Line Main | 126 |
| 15.4 Equations Environment | 128 |
| 16 Options | 132 |
| 16.1 Selection Tools | 132 |
| 16.2 Options Declarations | 132 |
| 16.3 Parameter Presets | 148 |
| 16.4 Component Selection | 150 |
| 16.5 Global and Package Options | 161 |

1 Implementation

This appendix documents the implementation for the various components of the eqnlines package.

The code for the package is based on the `amsmath` package, see the reference manual for details. 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.

2 General Support

In the following we describe general purpose supporting routines.

2.1 Debugging Messages

The package offers a verbose mode for debugging purposes. It outputs extra information on the current location within the code in order to track progress: **TODO:** describe

```

1 \def\eq@verbose@on{%
2   \def\eq@verbose@info##1{\PackageInfo{eqnlines}{##1}}%
3   \def\eq@verbose@infoarg##1##2{\eq@verbose@info{##1##2}}%
4 }
5 \def\eq@verbose@off{%
6   \let\eq@verbose@info\@gobble
7   \let\eq@verbose@infoarg\@gobbletwo
8 }
9 \eq@verbose@off

```

TODO: describe

```

10 \def\eq@verbose@msg@enterenv{entering \@currentenv}
11 \def\eq@verbose@msg@leaveenv{ leaving \@currentenv}
12 \def\eq@verbose@msg@start#1{starting \string#1}
13 \def\eq@verbose@msg@end#1{ \space ending \string#1}
14 \def\eq@verbose@msg@within#1{ \space within \string#1}
15 \def\eq@verbose@msg@enter#1{entering \string#1}
16 \def\eq@verbose@msg@leave#1{ leaving \string#1}
17 \def\eq@verbose@msg@startline@number{starting line \the\eq@row@}
18 \def\eq@verbose@msg@startline@new{starting new line}

```

2.2 Supporting Definitions

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

```

19 \let\eq@false\@undefined
20 \let\eq@true\@empty

```

TODO: for comparison within `\ifx`

```

21 \def\eq@relax{\relax}

```

TODO: describe

```

22 \def\eq@append#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
23 \def\eq@appendexpand#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
24 \def\eq@appendmacro#1#2{\eq@appendexpand#1{\unexpanded\expandafter{#2}}}
25 \def\eq@letcs#1{\expandafter\let\csname#1\endcsname}

```

2.3 Dollardollar Abstraction

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

```

26 \ifdefined\dollar@begin
27   \def\eq@dollar@begin{\dollar@begin}
28   \def\eq@dollar@end{\dollar@end}
29 \else
30   \def\eq@dollar@begin{$$}

```

```

31 \def\eqldollardollar@end{$$}
32 \fi

```

2.4 Look-Ahead in Alignment

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

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

```

33 \let\eq@ifnextchar@loose\kernel@ifnextchar
34 \long\def\eq@ifnextchar@tight#1#2#3{%
35   \let\reserved@d= #1%
36   \def\reserved@a{#2}%
37   \def\reserved@b{#3}%
38   \futurelet\@let@token\eq@ifnch@tight
39 }
40 \def\eq@ifnch@tight{%
41   \ifx\@let@token\reserved@d
42     \let\reserved@b\reserved@a
43   \fi
44   \reserved@b
45 }

```

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

```

46 \let\eq@atxi=@
47 \begingroup
48   \makeatother
49   \let\tmp=@%
50   \makeatletter
51   \global\let\eq@atxii\tmp
52 \endgroup

```

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

```

53 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
54 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
55 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
56 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
57 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{%
58   \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
59 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{%
60   \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}}

```

```

61 \long\def\eqL@testopt@loose#1#2{\eqL@ifnextchar@loose[{#1}{#1[{#2}]}]}%
62 \long\def\eqL@testopt@tight#1#2{\eqL@ifnextchar@tight[{#1}{#1[{#2}]}]}%
63 \long\def\eqL@teststaropt@loose#1#2#3{%
64   \eqL@ifstar@loose{\eqL@testopt@loose{#1}{#3}}{\eqL@testopt@loose{#2}{#3}}
65 \long\def\eqL@teststaropt@tight#1#2#3{%
66   \eqL@ifstar@tight{\eqL@testopt@tight{#1}{#3}}{\eqL@testopt@tight{#2}{#3}}
67 \long\def\eqL@teststaroropt@loose#1#2#3{%
68   \eqL@ifstar@loose{#1}{\eqL@testopt@loose{#2}{#3}}
69 \long\def\eqL@teststaroropt@tight#1#2#3{%
70   \eqL@ifstar@tight{#1}{\eqL@testopt@tight{#2}{#3}}
71 \long\def\eqL@gobbleopt[#1]{}
72 \long\def\eqL@gobbleoptone[#1]#2{}

```

TODO: describe

```

73 \def\eqL@testopt@default{\eqL@testopt@default}

```

TODO: describe

```

74 \let\eqL@parseopt@warn@env\@empty
75 \let\eqL@parseopt@warn@cr\@empty

```

TODO: describe

```

76 \def\eqL@parseopt@env{%
77   \let\eqL@parseopt@warn\eqL@parseopt@warn@env\eqL@parseopt}
78 \def\eqL@parseopt@cr{%
79   \let\eqL@parseopt@warn\eqL@parseopt@warn@cr\eqL@parseopt}

```

TODO: describe

```

80 \def\eqL@parseopt#1#2{%
81   \def\eqL@parseopt@case{#1}%
82   \def\eqL@parseopt@end{#2}%
83   \eqL@parseopt@peek
84 }
85 \def\eqL@parseopt@peek{%
86   \futurelet\eqL@parseopt@token\eqL@parseopt@select
87 }
88 \def\eqL@parseopt@select{%
89   \let\eqL@parseopt@next\eqL@parseopt@other
90   \ifx\eqL@parseopt@token\@sptoken
91     \let\eqL@parseopt@next\eqL@parseopt@end
92   \fi
93   \eqL@parseopt@case
94   \eqL@parseopt@next
95 }
96 \def\eqL@parseopt@other{\eqL@parseopt@warn\eqL@parseopt@end}
97 \def\eqL@parseopt@gobble#1{\eqL@parseopt@peek}

```

`\eqL@spbgroup` The second challenge is addressed by enclosing the look-ahead in spurious groups¹ which
`\eqL@spgroup` protect against triggering ‘&’. The macros `\eqL@spbgroup` and `\eqL@spgroup` 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 `\eqL@spgroup`:

```

98 \def\eqL@spbgroup{\iffalse{\fi\ifnum0=‘}\fi}

```

¹See <https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3040>,
<https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=amslatex/1834> and
<https://tex.stackexchange.com/questions/9897/showcase-of-brace-tricks-egroup-iffalse-fi-etc>.

```

99 \def\eql@speggroup{\ifnum0='{ \fi\iffalse}\fi}
100 \def\eql@srbgroup{\relax\iffalse{\fi\ifnum0='{ \fi}
101 \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:

```

102 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1{\eql@sregroup#2}}
103 \long\def\eql@ampprotecttwo#1#2#3{%
104   \eql@srbgroup#1{\eql@sregroup#2}{\eql@sregroup#3}}

```

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

```

105 \def\eql@ifnextchar@loose@ampsafe#1{%
106   \eql@ampprotecttwo{\eql@ifnextchar@loose#1}}
107 \def\eql@ifnextchar@tight@ampsafe#1{%
108   \eql@ampprotecttwo{\eql@ifnextchar@tight#1}}
109 \def\eql@ifstar@loose@ampsafe{\eql@ampprotecttwo\eql@ifstar@loose}
110 \def\eql@ifstar@tight@ampsafe{\eql@ampprotecttwo\eql@ifstar@tight}
111 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
112 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
113 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
114 \long\def\eql@teststaropt@tight@ampsafe{%
115   \eql@ampprotecttwo\eql@teststaropt@tight}

```

`\eql@amproof` We may want to replace L^AT_EX’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:

```

116 \let\eql@ifnextchar@org\@ifnextchar
117 \let\eql@ifstar@org\@ifstar
118 \let\eql@testopt@org\@testopt
119 \def\eql@amprevert{%
120   \let\@ifnextchar\eql@ifnextchar@org
121   \let\@testopt\eql@testopt@org
122   \let\@ifstar\eql@ifstar@org
123 }
124 \def\eql@amproof{%
125   \let\@ifnextchar\eql@ifnextchar@loose@ampsafe
126   \let\@testopt\eql@testopt@loose@ampsafe
127   \let\@ifstar\eql@ifstar@loose@ampsafe
128 }

```

2.5 Error Messages

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

```

129 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
130 \def\eql@warning{\PackageWarning{eqnlines}}

```

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

```

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

```

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

```

133 \def\eql@warn@tags@unused#1#2{\eql@warning{Unused equation #1:
134   #2 will be lost}}
135 \def\eql@warn@tags@multiple#1#2#3{\eql@warning{Multiple equation #1:
136   previous #2 will be lost#3}}
137 \def\eql@warn@label@unused{\eql@warn@tags@unused{\string\label}%
138   {label '\eql@tags@label'}}
139 \def\eql@warn@label@multiple#1{\eql@warn@tags@multiple{\string\label's}%
140   {label '\eql@tags@label'}{ and replaced by '#1'}}
141 \def\eql@warn@name@unused{\eql@warn@tags@unused{label name}%
142   {name declaration}}
143 \def\eql@warn@name@multiple{\eql@warn@tags@multiple{label names}%
144   {name declaration}{}}
145 \def\eql@warn@tag@unused{\eql@warn@tags@unused{\string>tag}%
146   {tag declaration}}
147 \def\eql@warn@tag@multiple{\eql@warn@tags@multiple{\string>tag's}%
148   {tag declaration will be lost}{}}
149 \def\eql@warn@ref@unused{\eql@warn@tags@unused{tag label}%
150   {tag label declaration}}
151 \def\eql@warn@ref@multiple{\eql@warn@tags@multiple{tag labels}%
152   {tag label declaration}{}}

153 \def\eql@warn@parseopt{\eql@warning{Unknown modifier token:
154   modifier parsing stopped}}
155 \def\eql@warn@parseopt@verbose{\eql@warning{Unknown modifier token:
156   '\meaning\eql@parseopt@token'}}

```

2.6 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/`:

```

157 \def\eql@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
158 \def\eql@amsmath@before#1{%
159   \@ifpackageloaded{amsmath}{\AddToHook{package/amsmath/before}{#1}}
160 \def\eql@amsmath@undefine#1{\eql@amsmath@before{\let#1\undefined}}
161 \def\eql@amsmath@let#1#2{\eql@amsmath@undefine#1\let#1#2}

```

TODO: temporary fix for development stages

```

162 \@ifpackageloaded{amsmath}{\%
163   \DeclareHookRule{package/amsmath/after}%
164   {eqnlines}{after}{latex-lab-testphase-math}}

```

2.7 PDF Tagging Support

`\eql@tagging@...` Proper PDF tagging² support requires a L^AT_EX version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with L^AT_EX versions around 2020: **TODO:** adjust to further developments

```

165 \let\eql@tagging@on\eql@false
166 \IfFormatAtLeastTF{2025-06-01}{%
167   \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}{}

```

²see <https://latex3.github.io/tagging-project/>


```

168 \ifdefined\eql@tagging@on
169   \def\eql@tagging@mathsave{%
170     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
171   \def\eql@tagging@mathaddlast{%
172     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
173   \def\eql@tagging@tagbegin{%
174     \UseTaggingSocket{math/display/tag/begin}}
175   \def\eql@tagging@tagend{%
176     \UseTaggingSocket{math/display/tag/end}}
177   \def\eql@tagging@tagsave{%
178     \UseTaggingSocket{math/luamml/mtable/tag/save}}
179   \def\eql@tagging@tagaddbox{%
180     \setbox\z@\copy\eql@tagbox%
181     \UseTaggingSocket{math/luamml/mtable/tag/set}}
182   \def\eql@tagging@tablesaverinner{%
183     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
184   \def\eql@tagging@tableaddinner{%
185     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
186   \def\eql@tagging@tablesavelines{%
187     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
188   \def\eql@tagging@tablesavealign{%
189     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
190   \def\eql@tagging@alignleft{%
191     \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
192   \def\eql@tagging@aligncenter{%
193     \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
194   \def\eql@tagging@alignright{%
195     \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:

```

196   \let\eql@single@doscan\eql@true
197   \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:

```

198   \def\eql@tagging@start{%
199     \csname bool_if:N\expandafter\endcsname
200       \csname l__math_collected_bool\endcsname{%
201         \edef\eql@tmp{\@currenvir}{\unexpanded\expandafter{\eql@tagging@opt}}%
202         \the\eql@scan@reg}%
203     \csname __math_process:nn\expandafter\endcsname\eql@tmp
204     \@kernel@math@registered@begin
205     \csname bool_set_true:N\expandafter\endcsname
206     \csname l__math_collected_bool\endcsname
207   }%
208 }
209 \def\eql@tagging@end{}
210 \def\eql@tagging@register@luamml#1{%
211   \AddToHook{package/luamml/after}{%
212     \eqletcs{c__luamml_label_#1_tl}{\empty}}
213 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}

```

When tagging is deactivated, provide empty definitions:

```

214 \else
215   \let\eql@tagging@mathsave\@empty
216   \let\eql@tagging@mathaddlast\@empty
217   \let\eql@tagging@tagbegin\@empty
218   \let\eql@tagging@tagend\@empty
219   \let\eql@tagging@tagsave\@empty
220   \let\eql@tagging@tagaddbox\@empty
221   \let\eql@tagging@tablesaveinner\@empty
222   \let\eql@tagging@tableaddinner\@empty
223   \let\eql@tagging@tablesavelines\@empty
224   \let\eql@tagging@tablesavealign\@empty
225   \let\eql@tagging@alignleft\@empty
226   \let\eql@tagging@aligncenter\@empty
227   \let\eql@tagging@alignright\@empty
228   \let\eql@tagging@start\@empty
229   \let\eql@tagging@end\@empty
230   \let\eql@tagging@register@luamml\@gobble
231   \let\eql@tagging@register@env\@gobble
232 \fi

```

2.8 Key-Value Processing

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

```

233 \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:

```

234 \def\eql@decide@relax{\eql@tmpb:=\relax}
235 \def\eql@decide@select#1#2#3#4{%
236   \def\eql@tmpa{#3}%
237   \let\eql@tmpd\@undefined
238   \@for\eql@tmpc=#4\do{%
239     \ifdefined\eql@tmpd\else
240       \edef\eql@tmpb{\noexpand\eql@tmpb:=\expandafter\@firstoftwo\eql@tmpc}%
241       \ifx\eql@tmpb\eql@decide@relax
242         \let\eql@tmpa\eql@relax
243       \fi
244       \expandafter\@for\eql@tmpb\do{%
245         \ifx\eql@tmpa\eql@tmpb
246           \edef\eql@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
247             \expandafter\@secondoftwo\eql@tmpc}}%
248         \fi

```

```

249     }%
250     \fi
251 }%
252 \ifdefined\eq1@tmpd
253     \eq1@tmpd
254 \else
255     \eq1@error{undefined value '#3' for option '#2' of '#1'}%
256 \fi
257 }

```

Decide between true and false or related pairs of values:

```

258 \def\eq1@decide@true{on,true,yes,enabled}
259 \def\eq1@decide@false{off,false,no,disabled}

```

`\eq1@decide@if`

```

260 \def\eq1@decide@if#1#2#3#4#5{%
261     \eq1@decide@select{#1}{#2}{#3}{%
262         {\eq1@decide@true{#4}},%
263         {\eq1@decide@false{#5}}}%

```

`\eq1@decide@bool` Store a boolean value into a conditional register:

```

264 \def\eq1@decide@bool#1#2#3#4{%
265     \eq1@decide@if{#1}{#2}{#3}{\let#4\eq1@true}{\let#4\eq1@false}}

```

Key Declaration.

`\eq1@define@key` For convenience, we define a wrapper for keyval's `\define@key` which accepts lists of categories and keys. We prepend the prefix `eq1@` to all our categories so that we can hide it from the user in error messages:

```

266 \def\eq1@define@key#1#2{%
267     \eq1@ifnextchar@loose[%
268         {\eq1@definekey@opt{#1}{#2}}%
269         {\eq1@definekey@noopt{#1}{#2}}%
270 }
271 \def\eq1@definekey@noopt#1#2#3{\eq1@definekey@for{#1}{#2}{#3}}
272 \def\eq1@definekey@opt#1#2[#3]#4{\eq1@definekey@for{#1}{#2}{[#3]{#4}}}
273 \def\eq1@definekey@for#1#2#3{%
274     \def\eq1@for@fn##1##2##3{\define@key{eq1@##3}{##2}{#3}}%
275     \edef\eq1@for@vara{\noexpand\eq1@for@vara:=#1}%
276     \expandafter\@for\eq1@for@vara\do{%
277         \edef\eq1@for@varb{\noexpand\eq1@for@varb:=#2}%
278         \expandafter\@for\eq1@for@varb\do{%
279             \edef\eq1@for@call##1{%
280                 \noexpand\eq1@for@fn{##1}{\eq1@for@varb}{\eq1@for@vara}}%
281                 \eq1@for@call{##1}%
282             }%
283         }%
284 }

```

`\eq1@setkeys` Our wrapper of keyval's `\setkeys` prepends the prefix `eq1@` to the category, and it expands the list argument once:

```

285 \def\eq1@setkeys#1#2{%
286     \def\eq1@tmp{\setkeys{eq1@#1}}%
287     \expandafter\eq1@tmp\expandafter{#2}%
288 }

```

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:

```
289 \let\eql@nextopt\@empty
290 \def\eql@nextopt@process#1{%
291   \eql@setkeys{#1}\eql@nextopt
292   \let\eql@tagging@opt\eql@nextopt
293   \global\let\eql@nextopt\@empty
294 }
```

`\eqnaddopt`

```
295 \newcommand{\eqnaddopt}[1]{%
296   \ifx\eql@nextopt\@empty
297     \eql@append\eql@nextopt{#1}%
298   \else
299     \eql@append\eql@nextopt{, #1}%
300   \fi
301 }
```

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

```
302 \newcommand{\eqnlineset}[1]{%
303   \eql@setkeys{setup}{#1}%
304   \ignorespaces
305 }
```

`\eql@control@default`

```
306 \protected\def\eql@control@default{%
307   \eql@warn@here\eqncontrol
308   \@gobble
309 }
310 \let\eqncontrol\eql@control@default
```

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

```
311 \newcommand{\eql@control}[1]{%
312   \relax
313   \eql@setkeys{control}{#1}%
314   \ignorespaces
315 }
```

2.9 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.

```
316 \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: **TODO**: may skip `\expandafter` before `\scantokens`?

```

317 \def\eql@scan@body@dump{\the\eql@scan@reg@}
318 \def\eql@scan@body@rescan{%
319   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
320 \let\eql@scan@body\eql@scan@body@dump

```

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

```

321 \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@`:

```

322 \def\eql@scan@env{%
323   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@env
324   \def\eql@scan@end{\expandafter\end\expandafter{\@currenvir}}}%
325   \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`:

```

326 \edef\eql@scan@iterate{\expandafter\noexpand\curname\@currenvir\endcurname}%
327 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
328 \ifdefined\eql@scan@par
329   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
330 \else
331   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
332 \fi
333 \eql@scan@iterate
334 }

```

`\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:

```

335 \long\def\eql@scan@env@iterate#1\end#2{%
336   \edef\eql@scan@stack{%
337     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
338   \ifx\@empty\eql@scan@stack
339     \@checkend{#2}%
340     \eql@scan@addto{#1}%

```

```

341 \expandafter\let\eql@scan@iterate\eql@scan@env@org
342 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@env
343 \expandafter\eql@scan@call
344 \else
345 \eql@scan@addto{#1\end{#2}}}%
346 \expandafter\eql@scan@iterate
347 \fi
348 }

```

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

```

349 \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.

```

350 \long\def\eql@scan@env@count#1\begin#2{%
351 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
352 }

```

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

```

353 \def\eql@scan@env@cancel{%
354 \@namedef{end\@currenvir}{\ignorespacesafterend}%
355 \eql@scan@end
356 }

```

Square Brackets. The following is a version of the above mechanism that scans for an equation body enclosed by `\[...]` paying attention to potential further instances of the square bracket enclosures contained in the body.

`\eql@scan@sqr` Start scanning for `\[`:

```

357 \def\eql@scan@sqr{%
358 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@sqr
359 \def\eql@scan@end{\}%
360 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
361 \let\eql@scan@sqr@org\[%\]
362 \ifdefined\eql@scan@par
363 \let\[\eql@scan@sqr@iterate%\]
364 \else
365 \let\[\eql@scan@sqr@iterate@nopar%\]
366 \fi
367 \[%\]
368 }

```

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

```

369 \long\def\eql@scan@sqr@iterate#1\[%\]{%
370 \edef\eql@scan@stack{%
371 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
372 \ifx\@empty\eql@scan@stack
373 \let\[\eql@scan@sqr@org%\]
374 \eql@scan@addto{#1}%
375 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@sqr
376 \expandafter\eql@scan@call
377 \else

```

```

378 \eql@scan@addto{#1\}}%
379 \expandafter\[%\]
380 \fi
381 }

```

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

```

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

```

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

```

383 \long\def\eql@scan@sqr@count#1\[#2{\% \]
384 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
385 }

```

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

```

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

```

Angle Brackets. The following is another version of the mechanism which scans for an equation body enclosed by `\<... \>`.

`\eql@scan@ang` Start scanning for `\>`:

```

387 \def\eql@scan@ang{%
388 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@ang
389 \def\eql@scan@end{\>}%
390 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
391 \let\eql@scan@ang@org\<\>
392 \ifdefined\eql@scan@par
393 \let\<\eql@scan@ang@iterate%\>
394 \else
395 \let\<\eql@scan@ang@iterate@nopar%\>
396 \fi
397 \<\>
398 }

```

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

```

399 \long\def\eql@scan@ang@iterate#1\>{%
400 \edef\eql@scan@stack{%
401 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
402 \ifx\@empty\eql@scan@stack
403 \let\<\eql@scan@ang@org%\>
404 \eql@scan@addto{#1}%
405 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@ang
406 \expandafter\eql@scan@call
407 \else
408 \eql@scan@addto{#1\>}%
409 \expandafter\<\>
410 \fi
411 }

```

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

```

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

```

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

```

413 \long\def\eql@scan@ang@count#1\<#2{%\>
414   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
415 }

```

Interface. TODO: describe

`\eql@sqr@open` If already in math mode, ignore the enclosed contents.

```

416 \protected\def\eql@sqr@open{%
417   \ifmmode
418     \expandafter\eql@sqr@cancel
419   \else
420     \expandafter\eql@equations@sqr@open
421   \fi
422 }

```

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

```

423 \protected\def\eql@sqr@close{%
424   \eql@error{'\string\]' may only close '\string\[']}%\]
425 }

```

`\eql@sqr@cancel`

```

426 \def\eql@sqr@cancel{%
427   \eql@error@mathmode{'\string\ [...\string\]]}%
428   \let\eql@scan@call\eql@scan@sqrang@cancel
429   \eql@scan@sqr
430 }

```

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

```

431 \protected\def\eql@ang@open{%
432   \ifmmode
433     \expandafter\eql@box@ang@open
434   \else
435     \expandafter\eql@equations@ang@open
436   \fi
437 }

```

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

```

438 \protected\def\eql@ang@close{%
439   \eql@error{'\string\>' may only close '\string\<']}%\>
440 }

```

3 Parameters and Registers

In the following, we collect parameter and register definitions.

3.1 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:

```
441 \let\eql@tagsleft\eql@false
```

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

```
442 \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`:

```
443 \def\eql@layoutleftmargin{\leftmargini}
444 \def\eql@layoutleftmarginmax{.5\maxdimen}
445 \def\eql@layoutleftmarginmin{\z@}
```

`\eql@mathstyle` The math style to be used within cells is specified by `\eql@mathstyle`:

```
446 \let\eql@mathstyle\displaystyle
```

`\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

```
447 \newdimen\eql@tagmargin@
448 \let\eql@tagmargin@val\@undefined
449 \newdimen\eql@tagmargin@ratio@
450 \eql@tagmargin@ratio@\p@
451 \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:

```
452 \newdimen\eql@indent@
453 \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:

```
454 \newdimen\eql@paddingleft@
455 \newdimen\eql@paddingright@
456 \let\eql@paddingleft@val\@undefined
457 \let\eql@paddingright@val\@undefined
```

`\eqldisplay@linewidth` **TODO:** describe

`\eqldisplay@marginleft`

`\eqldisplay@marginright`

```

458 \let\eqldisplay@linewidth\@undefined
459 \let\eqldisplay@marginleft\@undefined
460 \let\eqldisplay@marginright\@undefined

```

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

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

`\eql@box@condsep`

`\eql@box@matrixsep`

```

461 \def\eql@box@colsep{2em}
462 \def\eql@box@shortsep{1em}
463 \def\eql@box@condsep{\eql@box@shortsep}
464 \def\eql@box@matrixsep{\eql@box@shortsep}

```

`\eql@break@line@sep` **TODO:** describe

`\eql@break@line@shortsep`

`\eql@break@col@sep`

`\eql@break@col@shortsep`

```

465 \def\eql@break@line@sep{2em minus 1em}
466 \def\eql@break@line@shortsep{1em}
467 \def\eql@break@col@sep{2em minus 1em}
468 \def\eql@break@col@shortsep{1em}

```

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

```

469 \let\eql@spread@reset\eql@false
470 \def\eql@spread@val{\jot}
471 \newdimen\eql@spread@

```

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

```

472 \newdimen\eql@tagfuzz@
473 \eql@tagfuzz@16sp\relax

```

`\eqldisplay@height` An equation will appear to the surrounding text with a fixed apparent height and depth

`\eqldisplay@depth` specified by `\eqldisplay@height` and `\eqldisplay@depth`, respectively:

```

474 \def\eqldisplay@height\@undefined
475 \def\eqldisplay@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:

```

476 \def\eql@skip@mode@short{2}

477 \def\eql@skip@mode@cont@above{2}
478 \def\eql@skip@mode@cont@below{0}

479 \def\eql@skip@mode@par@above{3}
480 \def\eql@skip@mode@par@below{0}

481 \def\eql@skip@mode@top@above{4}
482 \def\eql@skip@mode@top@below{0}

483 \newcount\eql@skip@mode@leave@
484 \let\eql@skip@force@leave\@undefined

```

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

```

\eq@skip@force@below
\mode@above@ (counter) 485 \newcount\eq@skip@mode@above@
\mode@below@ (counter) 486 \newcount\eq@skip@mode@below@
487 \let\eq@skip@force@above@\undefined
488 \let\eq@skip@force@below@\undefined
489 \let\eq@skip@custom@above@\undefined
490 \let\eq@skip@custom@below@\undefined

```

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

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

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

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

```

491 \def\eq@skip@long@above{\abovedisplayskip}
492 \def\eq@skip@long@below{\belowdisplayskip}
493 \def\eq@skip@short@above{\abovedisplaysshortskip}
494 \def\eq@skip@short@below{\belowdisplaysshortskip}
495 \def\eq@skip@cont@above{\eq@skip@short@above}
496 \def\eq@skip@cont@below{\eq@skip@short@below}
497 \def\eq@skip@par@above{\eq@skip@long@above}
498 \def\eq@skip@par@below{\eq@skip@long@below}
499 \def\eq@skip@top@above{\eq@skip@long@above}
500 \def\eq@skip@top@below{\eq@skip@long@below}
501 \def\eq@skip@med@above{\abovedisplayskip/2}
502 \def\eq@skip@med@below{\belowdisplayskip/2}
503 \def\eq@skip@tag@above{\z@skip}
504 \def\eq@skip@tag@below{\z@skip}

```

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

```

505 \newdimen\eq@colsepmin@
506 \def\eq@colsepmin@val{1em}
507 \def\eq@colsepmax@val{.5\maxdimen}

```

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

```

508 \newdimen\eq@tagwidthmin@
509 \eq@tagwidthmin@\z@

```

`\eq@tagsepmin@ (dimen)` The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. T_EX's built-in value is half a quad³ in font number 2. As the tag is processed in text mode, we use 0.5em instead.

```

510 \newdimen\eq@tagsepmin@
511 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

³another half of a quad is left empty at the other end of the line.

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

```
\eql@equations@ang@opt
  \eql@box@ang@opt
512 \def\eql@equations@sqr@opt{equation}
513 \def\eql@equations@ang@opt{columns}
514 \def\eql@box@ang@opt{columns}
```

Multi-Line Align Mode.

```
515 \let\eql@columns@fulllength\eql@false
```

3.2 Registers

TODO: describe

General. **TODO:** describe

```
516 \newcount\eql@count@
517 \newdimen\eql@dimen@
518 \newskip\eql@skip@
```

TODO: describe

```
519 \let\eql@display@container\@empty
```

`\eql@cellbox@` (*box*) The box `\eql@cellbox@` holds the present alignment component and `\eql@tagbox@` the

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

`\eql@cellwidth@` (*dimen*) `\eql@tagwidth@` hold their widths. `\eql@prevwidth@` holds the width of the previous

`\eql@prevwidth@` (*dimen*) alignment component: **TODO:** adjust

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

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

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

```
520 \newbox\eql@cellbox@
```

```
521 \newbox\eql@tagbox@
```

```
522 \newdimen\eql@cellwidth@
```

```
523 \newdimen\eql@prevwidth@
```

```
524 \newdimen\eql@tagwidth@
```

```
525 \newdimen\eql@prevdepth@
```

```
526 \newcount\eql@prevgraf@
```

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

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

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

```
527 \newdimen\eql@totalwidth@
```

```
528 \newdimen\eql@tagwidth@max@
```

```
529 \newdimen\eql@totalheight@
```

```
530 \newdimen\eql@topheight@
```

```
531 \newdimen\eql@bottomdepth@
```

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

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

```
532 \newdimen\eql@line@height@
```

```
533 \newdimen\eql@line@depth@
```

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

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

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

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

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

`\eql@line@possep@` (*counter*)

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

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

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

```
534 \newdimen\eql@line@width@
```

```
535 \newdimen\eql@line@avail@
```

```
536 \newdimen\eql@line@pos@
```

```

537 \newcount\eql@line@availsep@
538 \newcount\eql@line@widthsep@
539 \newcount\eql@line@possep@
540 \newdimen\eql@line@offset@
541 \newdimen\eql@line@prevdepth@
542 \newdimen\eql@line@interline@

```

Rows and Columns.

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

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

```

543 \newcount\eql@row@
544 \newcount\eql@totalrows@
545 \newcount\eql@tagrows@

```

`\eql@column@`

`\eql@totalcolumns@`

```

546 \newcount\eql@column@
547 \newcount\eql@totalcolumns@

```

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

```

548 \newdimen\eql@colsep@

```

`\intercolumns@` (*counter*)

```

549 \newcount\eql@intercolumns@

```

Vertical Spacing Adjustments.

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

```

550 \newdimen\eql@display@firstavail@
551 \def\eql@display@firstavail@set#1{%
552   \ifnum\eql@row@=\@ne
553     \global\eql@appendexpand\eql@display@container{%
554       \eql@display@firstavail@\the#1\relax}%
555   \fi
556 }

```

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

```

557 \newdimen\eql@display@aboveextend@
558 \newdimen\eql@display@belowextend@

```

Shared Registers.

`\ifmeasuring@` (*bool*) All display environments get typeset twice – once during a “measuring” phase and then again during a “production” phase. We reuse the original `amsmath` definition

`\ifmeasuring@` to determine which case we're in, so we and other packages may take appropriate action. It does not hurt to define this conditional in any case. We should tell `hyperref` about measuring processes as we're not `amsmath` and not being catered for:

```
559 \ifdefined\measuring@true\else
560   \expandafter\newif\csname ifmeasuring@\endcsname
561 \fi
562 \AddToHook{package/hyperref/after}{%
563   \ifdefined\Hy@ifnotmeasuring
564     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}%
565   \fi
566 }
```

`\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:

```
567 \ifdefined\@displaytrue\else
568   \expandafter\newif\csname if@display\endcsname
569   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
570 \fi
```

3.3 Hooks

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

```
571 \let\eql@hook@blockbefore\@empty
572 \let\eql@hook@blockafter\@empty
573 \let\eql@hook@blockin\@empty
574 \let\eql@hook@blockout\@empty
575 \let\eql@hook@linein\@empty
576 \let\eql@hook@lineout\@empty
577 \let\eql@hook@colin\@empty
578 \let\eql@hook@colout\@empty
579 \let\eql@hook@eqin\@empty
580 \let\eql@hook@eqout\@empty
581 \let\eql@hook@innerleft\@empty
582 \let\eql@hook@innerright\@empty
583 \let\eql@hook@innerlead\@empty
```

4 Features

4.1 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. An undefined value indicates that the punctuation should be handed down to the next lower level:

`\eql@punct@line`

`\eql@punct@block` **TODO:** update

`\eql@punct@next`

`\eql@punct@top` 584 `\let\eql@punct@col\@empty`

`\eql@punct@cases`

```

585 \let\eqlopunct@line\@undefined
586 \let\eqlopunct@block\@undefined
587 \let\eqlopunct@next\@undefined
588 \let\eqlopunct@top\@undefined
589 \let\eqlopunct@cases\@empty

```

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

```

590 \let\eqlopunct@class\@empty
591 \let\eqlopunct@sep\@empty

```

`\eqlopunct@top@set` **TODO:** describe

```

\eqlopunct@top@stop
\eqlopunct@top@reset
592 \def\eqlopunct@top@set{\let\eqlopunct@top\eqlopunct@block}
593 \def\eqlopunct@top@stop{\let\eqlopunct@top\relax}
594 \def\eqlopunct@top@reset{\let\eqlopunct@top\@undefined}

```

`\eqlopunct@set` **TODO:** describe

```

595 \def\eqlopunct@tilde{~}
596 \def\eqlopunct@set#1#2{%
597   \def#1{#2}%
598   \ifx#1\eqlopunct@relax
599     \let#1\@undefined
600   \fi
601   \ifx#1\eqlopunct@tilde
602     \let#1\@empty
603   \fi
604 }
605 \def\eqlopunct@clear{%
606   \let\eqlopunct@col\@empty
607   \let\eqlopunct@line\@empty
608   \let\eqlopunct@block\@empty
609 }
610 \def\eqlopunct@next@clear{\let\eqlopunct@next\@empty}

```

Set the punctuation for 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:

TODO: describe

```

611 \def\eqlopunct@adopt{%
612   \eqlopunct@ifstar@tight\eqlopunct@adopt@relax\eqlopunct@adopt@set}
613 \def\eqlopunct@adopt@set#1{\eqnadopt{punct={#1}}\ignorespaces}
614 \def\eqlopunct@adopt@relax{\eqnadopt{punct*}\ignorespaces}

```

TODO: describe

```

615 \def\eqlopunct@setnext{%
616   \eqlopunct@ifstar@tight\eqlopunct@setnext@relax\eqlopunct@setnext@set}
617 \def\eqlopunct@setnext@set#1{%
618   \eqlopunct@set\eqlopunct@next{#1}%
619   \ifdefined\eqlopunct@next\else\let\eqlopunct@next\relax\fi
620   \ignorespaces}
621 \def\eqlopunct@setnext@relax{\let\eqlopunct@next\relax\ignorespaces}

```

`\eqnpunct` **TODO:** describe

```

622 \let\eqnpunct\eqlopunct@adopt

```

`\eql@punct@fill@next` Fill the next punctuation:

```

623 \def\eql@punct@fill@next#1{%
624   \ifdefined\eql@punct@next
625     \ifx\eql@punct@next\relax
626       \let\eql@punct@next\undefined
627     \fi
628   \else
629     \ifx\eql@punct@top\relax\else
630       \let\eql@punct@next#1%
631     \fi
632   \fi
633 }
```

`eql@punct@output@next` Output the next punctuation. If non-empty, prepend some separation:

```

634 \def\eql@punct@output@next{%
635   \ifx\eql@punct@next\@empty\else
636     \ifmmode\eql@punct@class\fi
637     \eql@punct@sep
638     \eql@punct@next
639   \fi
640   \let\eql@punct@next\undefined
641 }
```

`eql@punct@print@next` Print the next punctuation if available:

```

642 \def\eql@punct@print@next{%
643   \ifdefined\eql@punct@next
644     \eql@punct@output@next
645   \fi
646 }
```

`eql@punct@apply@next` Print the next punctuation if available. Stop further punctuation within the current group:

```

647 \def\eql@punct@apply@next{%
648   \ifdefined\eql@punct@next
649     \eql@punct@output@next
650     \eql@punct@top@stop
651   \fi
652 }
```

`\eql@punct@print@col` Print the punctuation for the present column:

```

653 \def\eql@punct@print@col{%
654   \eql@punct@fill@next\eql@punct@col
655   \eql@punct@print@next
656 }
```

`\eql@punct@apply@col` Output the punctuation for the present column. Stop further punctuation within the current group:

```

657 \def\eql@punct@apply@col{%
658   \eql@punct@fill@next\eql@punct@col
659   \eql@punct@apply@next
660 }
```

Output the punctuation for the present line unless disabled:

`\eqlopunct@print@line`

```
661 \def\eqlopunct@print@line{%
662   \eqlopunct@fill@next\eqlopunct@line
663   \eqlopunct@print@next
664 }
```

Output the punctuation for the present line unless disabled. Stop further punctuation within the current group:

`\eqlopunct@apply@line`

```
665 \def\eqlopunct@apply@line{%
666   \eqlopunct@fill@next\eqlopunct@line
667   \eqlopunct@apply@next
668 }
```

`\eqlopunct@apply@block` Output the punctuation for the present block unless disabled. Stop further punctuation within the current group:

```
669 \def\eqlopunct@apply@block{%
670   \eqlopunct@fill@next\eqlopunct@block
671   \eqlopunct@apply@next
672 }
```

`\eqlopunct@apply@top` Output the top punctuation unless disabled. Stop further punctuation globally:

```
673 \def\eqlopunct@apply@top{%
674   \eqlopunct@fill@next\eqlopunct@top
675   \eqlopunct@print@next
676   \global\eqlopunct@top@stop
677 }
```

`\eqnpunctapply` Output the top punctuation unless disabled. Stop further punctuation globally:

```
678 \newcommand{\eqnpunctapply}{\ifmmode\else\unskip\fi\eqlopunct@apply@top}
```

4.2 Math Classes at Alignment

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

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

```
679 \def\eqlopclass@innerright@sel@{%
680   \ifdim\eqlop@prevwidth@=\z@
681     \eqlopclass@innerlead
682   \else
683     \eqlopclass@innerright
684   \fi
685 }
```

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

```
686 \def\eqlopclass@innerleft@set#1{%
687   \def\eqlopclass@innerleft@{#1}%
688 }
```

```

689 \def\eq@class@innerright@set#1{%
690   \def\eq@class@innerright{#1}%
691   \let\eq@class@innerright@sel\eq@class@innerright
692 }
693 \def\eq@class@innerlead@set#1{%
694   \def\eq@class@innerlead{#1}%
695   \let\eq@class@innerright@sel\eq@class@innerright@sel@
696 }

697 \def\eq@class@rel@symb{=}
698 \def\eq@class@rel@break#1{\eq@class@rel@start{#1}\mathclose{}}
699 \def\eq@class@rel@start#1{\mathrel{\phantom{#1}}}
700 \def\eq@class@rel@break@set#1{\def\eq@class@rel@break##1{#1}}
701 \def\eq@class@rel@start@set#1{\def\eq@class@rel@start##1{#1}}
702 \def\eq@class@rel@break@skip{\thickmuskip-\medmuskip}
703 \def\eq@class@rel@amp{&}
704 \def\eq@class@rel@amprelax{&\relax}
705 \def\eq@class@rel@relaxamp{\relax&}
706 \def\eq@class@rel@ordamp{{}&}
707 \def\eq@class@rel@make#1{%
708   \ifdefined\eq@class@rel@after
709     \def\eq@tmp{&#1}%
710     \ifx\eq@tmp\eq@class@rel@amp
711       \def\eq@tmp{&\eq@class@rel@break\eq@class@rel@symb
712         \mskip\muexpr\eq@class@rel@break@skip\relax}%
713     \else\ifx\eq@tmp\eq@class@rel@amprelax
714       \def\eq@tmp{&\eq@class@rel@start\eq@class@rel@symb}%
715     \fi\fi
716   \else
717     \def\eq@tmp{#1&}%
718     \ifx\eq@tmp\eq@class@rel@amp
719       \def\eq@tmp{&\mathclose{}}%
720       \mskip\muexpr\eq@class@rel@break@skip\relax}%
721     \fi
722     \ifx\eq@tmp\eq@class@rel@relaxamp
723       \def\eq@tmp{&\mathclose{}\mathopen{}\mathrel{}}%
724     \fi
725     \ifx\eq@tmp\eq@class@rel@ordamp
726       \def\eq@tmp{&\mathclose{}}}%
727     \fi
728   \fi
729   \eq@tmp
730 }

```

`\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`):

```

731 \def\eq@class@ampeq{%
732   \eq@class@innerleft@set{}%
733   \eq@class@innerright@set{}}%
734   \let\eq@class@rel@after\eq@true
735 }
736 \def\eq@class@eqamp{%
737   \eq@class@innerleft@set{}%
738   \eq@class@innerright@set{\mathrel{}}%
739   \eq@class@innerlead@set{\mathclose{}}%
740   \let\eq@class@rel@after\eq@false
741 }
742 \eq@class@ampeq

```

4.3 Framed Cells

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

```
743 \let\eql@frame@cmd\@undefined
744 \newdimen\eql@frame@margin@
745 \def\eql@frame@set[#1]{%
746   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
747   \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbx}
748   \def\eql@frame@measure{%
749     \setbox\z@\hbox{\eql@frame@cmd}}%
750     \eql@frame@margin@.5\wd\z@
751 }
752 \def\eql@frame@print{%
753   \setbox\eql@cellbox@\hbox{%
754     \eql@frame@cmd{\unhbox\eql@cellbox@}%
755   }%
756 }
757 \def\eql@frame@adjust{%
758   \setbox\eql@cellbox@\hbox{%
759     \eql@frame@cmd{%
760       \unhbox\eql@cellbox@
761       \unkern
762       \unskip
763     }%
764     \hfil
765     \kern\z@
766   }%
767 }
```

4.4 Single-Line Composition

TODO: describe

```
\eql@break@line
\eql@break@col
768 \def\eql@break@line{%
769   \let\eql@break@sep\eql@break@line@sep
770   \let\eql@break@shortsep\eql@break@line@shortsep
771   \let\eql@break@print\eql@punct@print@line
772   \let\eql@punct@term\eql@false
773   \let\eql@class@rel@composed\@undefined
774   \eql@ampprotect\eql@break@test\eql@break@process}
775 \def\eql@break@col{%
776   \let\eql@break@sep\eql@break@col@sep
777   \let\eql@break@shortsep\eql@break@col@shortsep
778   \let\eql@break@print\eql@punct@print@col
779   \let\eql@punct@term\eql@false
780   \let\eql@class@rel@composed\@undefined
781   \eql@ampprotect\eql@break@test\eql@break@process}
782 \def\eql@break@cr{%
783   \let\eql@break@sep\eql@break@line@sep
784   \let\eql@break@shortsep\eql@break@line@shortsep
785   \let\eql@break@print\eql@punct@print@line
786   \let\eql@punct@term\eql@false
787   \let\eql@class@rel@composed\@undefined
788   \eql@ampprotect\eql@break@cr@test\eql@break@process}
789 \def\eql@break@amp{%
```

```

790 \eql@ampprotecttwo\eql@break@amp@testescape
791 \eql@amp@org\eql@break@process}
792 \def\eql@break@amp@testescape#1#2{%
793 \eql@ifnextgobble@tight/{#1}{\eql@break@amp@testnoescape{#2}}}
794 \def\eql@break@amp@testnoescape#1{%
795 \relax
796 \let\eql@break@sep\eql@break@col@sep
797 \let\eql@break@shortsep\eql@break@col@shortsep
798 \let\eql@break@print\eql@punct@print@col
799 \let\eql@punct@term\eql@false
800 \let\eql@class@rel@composed\@undefined
801 \eql@break@amp@test{#1}}

```

TODO: describe

```

802 \def\eql@break@test@setopt{\let\eql@break@test\eql@break@testopt}
803 \def\eql@break@test@setall{\let\eql@break@test\eql@break@testall}
804 \def\eql@break@cr@test@setopt{\let\eql@break@cr@test\eql@break@testopt}
805 \def\eql@break@cr@test@setall{\let\eql@break@cr@test\eql@break@testall}
806 \def\eql@break@amp@test@setopt{\let\eql@break@amp@test\eql@break@testopt}
807 \def\eql@break@amp@test@setall{\let\eql@break@amp@test\eql@break@testall}

```

\eql@break@testopt **TODO:** describe

```

808 \def\eql@break@testopt#1{\eql@ifstar@tight
809 {\let\eql@break@sep\eql@break@shortsep#1}{\eql@break@testopt@arg{#1}}}
810 \def\eql@break@testopt@arg#1{\eql@ifnextchar@tight[%
811 {\eql@break@testopt@set{#1}}{#1}}
812 \def\eql@break@testopt@set#1[#2]{\def\eql@break@sep{#2}#1}

```

\eql@break@testall **TODO:** describe
@break@testall@parse

```

813 \def\eql@break@testall{\eql@parseopt@cr\eql@break@testall@parse}
814 \def\eql@break@testall@parse{%
815 \ifx\eql@parseopt@token[%
816 \let\eql@parseopt@next\eql@break@parse@val
817 \fi
818 \ifx\eql@parseopt@token*%
819 \let\eql@parseopt@next\eql@break@parse@star
820 \fi
821 \ifx\eql@parseopt@token.%
822 \let\eql@parseopt@next\eql@parseopt@punctpass
823 \fi
824 \ifx\eql@parseopt@token,%
825 \let\eql@parseopt@next\eql@parseopt@punctpass
826 \fi
827 \ifx\eql@parseopt@token~%
828 \let\eql@parseopt@next\eql@parseopt@punctpass
829 \fi
830 \ifx\eql@parseopt@token'%
831 \let\eql@parseopt@next\eql@parseopt@punctnext
832 \fi
833 \ifx\eql@parseopt@token!%
834 \let\eql@parseopt@next\eql@parseopt@punctterm
835 \fi
836 \ifx\eql@parseopt@token/%
837 \let\eql@parseopt@next\eql@parseopt@punctclear
838 \fi
839 \ifx\eql@parseopt@token=%

```

```

840 \let\eql@parseopt@next\eql@parseopt@relsyb
841 \fi
842 \ifx\eql@parseopt@token;%
843 \let\eql@parseopt@next\eql@parseopt@relcont
844 \fi
845 \ifx\eql@parseopt@token:%
846 \let\eql@parseopt@next\eql@parseopt@relstart
847 \fi
848 \ifx\eql@parseopt@token|%
849 \let\eql@parseopt@next\eql@parseopt@relord
850 \fi
851 \ifx\eql@parseopt@token?%
852 \let\eql@parseopt@next\eql@break@parse@rel
853 \fi
854 }
855 \def\eql@break@parse@val[#1]{%
856 \def\eql@break@sep{#1}\eql@parseopt@peek}
857 \def\eql@break@parse@star#1{%
858 \let\eql@break@sep\eql@break@shortsep\eql@parseopt@peek}
859 \def\eql@break@parse@rel#1#2{%
860 \def\eql@class@rel@composed{#2}\eql@parseopt@end}

```

`\eql@break@process`

```

861 \def\eql@break@process{%
862 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
863 \ifdefined\eql@class@rel@composed
864 \eql@class@rel@composed
865 \else
866 \eql@break@print
867 \hspace{\glueexpr\eql@break@sep\relax}%
868 \fi
869 }

```

`\eql@break@join`

```

870 \def\eql@break@join{\eql@srbgroup
871 \eql@ifstar@tight
872 {\eql@break@join@opt[\eql@break@col@shortsep]}%
873 {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
874 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
875 \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

`\eqnsep` **TODO:** expand to lines and columns mode

`\eqnbreak`

`\eqnjoin`

```

876 \newcommand{\eqnsep}{\eql@break@col}
877 \newcommand{\eqnbreak}{\eql@break@line}
878 \newcommand{\eqnjoin}{\eql@break@join}

```

4.5 Alternative Content Description

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

```

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

```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```
880 \let\eq\@tags@autolabel\eq@false
881 \let\eq\@tags@autotag\eq@true
882 \let\eq\@tags@warn\eq@true

883 \def\eq\@tags@name@generic{[equation]}

884 \let\eq\@tagpos@doconvert\eq@false

885 \def\eq\@tagpos@snap{4pt}
```

Registers.

```
886 \let\eq\@numbering@mode\@undefined

887 \let\eq\@numbering@active\eq@false
888 \let\eq\@numbering@multi\eq@true

889 \let\eq\@tags@container\@undefined
890 \def\eq\@tags@container@clear{%
891   \let\eq\@tags@label\@undefined
892   \let\eq\@tags@name\@undefined
893   \let\eq\@tags@tag\@undefined
894   \let\eq\@tags@ref\@undefined
895   \let\eq\@tags@anchor\@empty
896   \eq\@tagpos@shift@\z@
897   \eq\@tagpos@smashup@\z@
898   \eq\@tagpos@smashdown@\z@
899   \let\eq\@tagpos@reserve\eq@true
900 }

901 \let\eq\@tags@label\@undefined
902 \let\eq\@tags@name\@undefined
903 \let\eq\@tags@tag\@undefined
904 \let\eq\@tags@ref\@undefined
905 \let\eq\@tags@frame@cmd\@firstofone

\eq\@tags@glabel@ (counter)

906 \newcount\eq\@tags@glabel@
907 \eq\@tags@glabel@\z@
908 \def\eq\@tags@glabel{equation.\eq-\the\eq\@tags@glabel@}
909 \def\eq\@tags@glabel@step{\global\advance\eq\@tags@glabel@\@ne}

910 \let\eq\@tagpos@continuous\eq@false

911 \newcount\eq\@tagpos@row@
912 \newcount\eq\@tagpos@prevrow@
913 \newdimen\eq\@tagpos@shift@
914 \newdimen\eq\@tagpos@smashup@
915 \newdimen\eq\@tagpos@smashdown@
```

```

916 \newdimen\eql@tagpos@current@
917 \newdimen\eql@tagpos@plain@
918 \newdimen\eql@tagpos@raised@
919 \newdimen\eql@tagpos@target@
920 \newdimen\eql@tagpos@headroom@
921 \newdimen\eql@tagpos@footroom@

```

5.2 Schemes

TODO: describe

Table.

```

922 \def\eql@numbering@tab@sub{sub}
923 \def\eql@numbering@tab@all{all}
924 \def\eql@numbering@tab@first{first}
925 \def\eql@numbering@tab@last{last}
926 \def\eql@numbering@tab@in{in}
927 \def\eql@numbering@tab@out{out}
928 \def\eql@numbering@tab@middle{middle}
929 \def\eql@numbering@tab@best{best}
930 \def\eql@numbering@tab@here{here}
931 \def\eql@numbering@tab@top{top}
932 \def\eql@numbering@tab@bottom{bottom}
933 \def\eql@numbering@tab@center{center}
934 \def\eql@numbering@tab@centerone{centerone}
935 \def\eql@numbering@tab@median{median}
936 \def\eql@numbering@tab@baseline{baseline}

937 \let\eql@numbering@mode\eql@numbering@tab@all
938 \let\eql@numbering@mode@multi\eql@numbering@tab@all
939 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```

940 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
941 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
942 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
943 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
944 \let\eql@numbering@tab@outside\eql@numbering@tab@out
945 \let\eql@numbering@tab@inside\eql@numbering@tab@in
946 \let\eql@numbering@tab@within\eql@numbering@tab@in
947 \let\eql@numbering@tab@opt\eql@numbering@tab@best
948 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
949 \let\eql@numbering@tab@pick\eql@numbering@tab@here
950 \let\eql@numbering@tab@med\eql@numbering@tab@median
951 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
952 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone

```

TODO: describe

```

953 \let\eql@numbering@tab@a\eql@numbering@tab@all
954 \let\eql@numbering@tab@s\eql@numbering@tab@sub
955 \let\eql@numbering@tab@f\eql@numbering@tab@first
956 \let\eql@numbering@tab@l\eql@numbering@tab@last
957 \let\eql@numbering@tab@o\eql@numbering@tab@out
958 \let\eql@numbering@tab@i\eql@numbering@tab@in
959 \let\eql@numbering@tab@h\eql@numbering@tab@here

```

```

960 \let\eql@numbering@tab@t\eql@numbering@tab@top
961 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
962 \let\eql@numbering@tab@c\eql@numbering@tab@center
963 \let\eql@numbering@tab@m\eql@numbering@tab@median
964 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
965 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
966 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
967 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

968 \def\eql@numbering@init@all{\let\eql@numbering@multi\eql@true}

```

TODO: describe

```

969 \def\eql@numbering@init@sub{%
970   \let\eql@numbering@multi\eql@true
971   \ifdefined\eql@subequations@active
972     \let\eql@numbering@mode\eql@numbering@tab@all
973   \else
974     \let\eql@numbering@subeq@use\eql@true
975   \fi
976 }

977 \def\eql@numbering@init@first{\eql@tagpos@row@{\@ne}
978 \def\eql@numbering@init@last{\eql@tagpos@row@{\@MM}
979 \def\eql@numbering@init@here{\eql@tagpos@row@{\m@ne}

```

TODO: describe

```

980 \def\eql@numbering@init@in{%
981   \ifdefined\eql@tagsleft
982     \eql@numbering@init@last
983   \else
984     \eql@numbering@init@first
985   \fi
986 }

```

TODO: describe

```

987 \def\eql@numbering@init@out{%
988   \ifdefined\eql@tagsleft
989     \eql@numbering@init@first
990   \else
991     \eql@numbering@init@last
992   \fi
993 }

```

TODO: describe

```

994 \def\eql@tagpos@eval@middle{%
995   \ifnum\eql@tagpos@row@=\z@
996     \eql@tagpos@row@\numexpr(\eql@totalrows@
997       +\ifdefined\eql@tagsleft\z@\else\@ne\fi)/\tw@\relax
998   \fi
999 }

```

TODO: describe

```

1000 \def\eql@tagpos@eval@best{%
1001   \ifnum\eql@tagpos@row@=\z@

```



```

1002 \let\eql@numbering@best@use\eql@true
1003 \eql@numbering@init@out
1004 \fi
1005 }

```

TODO: describe

```

1006 \def\eql@numbering@init@continuous{\let\eql@tagpos@continuous\eql@true}

```

TODO: describe

```

1007 \let\eql@numbering@init@top\eql@numbering@init@continuous
1008 \def\eql@tagpos@eval@top{%
1009 \eql@tagpos@current@z@
1010 }

```

TODO: describe

```

1011 \let\eql@numbering@init@bottom\eql@numbering@init@continuous
1012 \def\eql@tagpos@eval@bottom{%
1013 \eql@tagpos@current@\dimexpr\eql@totalheight@
1014 -\eql@tagheight@block@-\eql@tagdepth@block@\relax
1015 }

```

TODO: describe

```

1016 \let\eql@numbering@init@center\eql@numbering@init@continuous
1017 \def\eql@tagpos@eval@center{%
1018 \ifnum\eql@totalrows@=\@ne
1019 \eql@tagpos@row@\@ne
1020 \fi
1021 \eql@tagpos@current@\dimexpr(\eql@totalheight@
1022 -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
1023 }

```

TODO: describe

```

1024 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
1025 \def\eql@tagpos@eval@centerone{%
1026 \eql@tagpos@current@\dimexpr(\eql@totalheight@
1027 -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
1028 }

```

TODO: describe

```

1029 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
1030 \def\eql@tagpos@eval@baseline{%
1031 \eql@tagpos@current@\dimexpr(\eql@totalheight@
1032 +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
1033 }

```

TODO: describe

```

1034 \let\eql@numbering@init@median\eql@numbering@init@continuous
1035 \def\eql@tagpos@eval@median{%
1036 \ifnum\eql@tagpos@row@=\z@
1037 \ifodd\eql@totalrows@
1038 \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
1039 \else
1040 \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
1041 \eql@dimensions@get\eql@tagpos@row@
1042 \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
1043 +(\eql@line@interline@-\eql@tagheight@block@

```

```

1044      +\eql@tagdepth@block@)/\tw@/relax
1045    \fi
1046    \ifnum\eql@totalrows@=\@ne
1047      \eql@tagpos@row@\@ne
1048    \else
1049      \eql@tagpos@adjust@eval@convert
1050      \eql@tagpos@row@\z@
1051    \fi
1052  \fi
1053 }

```

Selection.

```

1054 \def\eql@numbering@set#1{%
1055   \ifcsname eql@numbering@tab@#1\endcsname
1056     \expandafter\let\expandafter\eql@numbering@mode
1057     \csname eql@numbering@tab@#1\endcsname
1058   \ifx\eql@numbering@mode\eql@numbering@tab@all
1059     \let\eql@numbering@mode@multi\eql@numbering@mode
1060   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
1061     \let\eql@numbering@mode@multi\eql@numbering@mode
1062   \else
1063     \let\eql@numbering@mode@single\eql@numbering@mode
1064   \fi\fi
1065 \else
1066   \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
1067   \let\eql@numbering@mode\eql@numbering@tab@all
1068 \fi
1069 }

```

TODO: describe

```

1070 \def\eql@numbering@init{%
1071   \let\eql@numbering@multi\eql@false
1072   \let\eql@tagpos@continuous\eql@false
1073   \let\eql@numbering@subeq@use\eql@false
1074   \let\eql@numbering@best@use\eql@false
1075   \eql@tagpos@row@\z@
1076   \csname eql@numbering@init@\eql@numbering@mode\endcsname
1077   \ifdefined\eql@numbering@active
1078     \let\eql@numbering@eqnswinit@eqnswtrue
1079   \else
1080     \let\eql@numbering@eqnswinit@eqnswfalse
1081   \fi
1082   \let\eql@numbering@active\eql@false
1083 }

```

5.3 Interface

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

```

1084 \eql@amsmath@after{%
1085   \let\eql@print@eqnum@default\print@eqnum
1086   \let\eql@incr@eqnum@default\incr@eqnum
1087 }

```

TODO: describe

```

1088 \protected\def\donumber{%
1089   \if@eqnsw\else
1090     \global\@eqnswtrue
1091     \ifx\print@eqn\@empty
1092       \global\let\print@eqn\eq\print@eqnum@default
1093     \fi
1094     \ifx\incr@eqn\@empty
1095       \global\let\incr@eqn\eq\incr@eqnum@default
1096     \fi
1097   \fi
1098 }

```

TODO: reconsider operation

\numberhere

```

1099 \protected\def\eq\@numberhere{%
1100   \ifdefined\eq\@numbering@multi
1101     \global\@eqnswtrue
1102   \else
1103     \global\eq\@tagpos@row@\eq\@row@
1104   \fi
1105 }

```

TODO: describe

\numbernext

```

1106 \protected\def\eq\@numbernext{%
1107   \ifdefined\eq\@numbering@multi
1108     \global\@eqnswfalse
1109   \else
1110     \ifdefined\eq\@tagpos@continuous\else
1111       \ifnum\eq\@tagpos@row@=\eq\@row@
1112         \global\advance\eq\@tagpos@row@\@ne
1113       \fi
1114     \fi
1115   \fi
1116 }

```

Activation Trigger.

```

1117 \def\eq\@tags@autoenable{%
1118   \global\@eqnswtrue
1119   \ifnum\eq\@tagpos@row@=\m@ne
1120     \numberhere
1121   \fi
1122 }

```

Labels. **TODO:** describe

\eq\@label@org

```

1123 \let\eq\@label@org\label

```

TODO: describe

```

1124 \def\eq\@label@gobble{\eq\@ampprotect\eq\@testopt@tight\eq\@gobbleoptone{}}

```

TODO: describe

```
1125 \protected\def\eql@label{%
1126   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
1127 }
```

TODO: describe

```
1128 \def\eql@tags@add@labelname[#1]#2{%
1129   \def\eql@tmp{#1}%
1130   \ifx\eql@tmp\eql@testopt@default\else
1131     \eql@tags@add@name{#1}%
1132   \fi
1133   \eql@tags@add@label{#2}%
1134 }
```

TODO: describe

```
1135 \def\eql@tags@set@label#1{%
1136   \ifdefined\eql@tags@warn
1137     \ifdefined\eql@tags@label
1138       \eql@warn@label@multiple{#1}%
1139     \fi
1140   \fi
1141   \def\eql@tags@label{#1}%
1142 }
```

TODO: describe

```
1143 \def\eql@tags@set@name#1{%
1144   \ifdefined\eql@tags@warn
1145     \ifdefined\eql@tags@name
1146       \eql@warn@name@multiple
1147     \fi
1148   \fi
1149   \def\eql@tags@name{#1}%
1150 }
```

TODO: describe

```
1151 \def\eql@tags@add@label#1{%
1152   \ifdefined\eql@tags@autolabel
1153     \eql@tags@autoenable
1154   \fi
1155   \global\eql@appendexpand\eql@tags@container{%
1156     \noexpand\eql@tags@set@label{#1}}%
1157 }
```

TODO: describe

```
1158 \def\eql@tags@add@name#1{%
1159   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1160   \global\eql@appendmacro\eql@tags@container\eql@tmp
1161 }
```

TODO: describe

```
1162 \def\eql@tags@addblock@label#1{%
1163   \eql@appendexpand\eql@tags@container@block{%
1164     \noexpand\eql@tags@set@label{#1}}%
1165 }
```

TODO: describe

```

1166 \def\eql@tags@addblock@name#1{%
1167   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1168   \eql@appendmacro\eql@tags@container@block\eql@tmp
1169 }

```

Tags. **TODO:** describe

`\eql@tag@default`

```

1170 \protected\def\eql@tag@default{%
1171   \eql@warn@here\tag
1172   \eql@tag@gobble
1173 }
1174 \let\tag\eql@tag@default

```

`\eql@tag@gobble`

```

1175 \def\eql@tag@gobble{%
1176   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}

```

TODO: describe

```

1177 \protected\def\eql@tag{%
1178   \eql@ampprotecttwo\eql@teststaropt@tight
1179   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}%
1180   \eql@testopt@default
1181 }

```

`\eql@tags@add@tagref`

```

1182 \def\eql@tags@add@tagref[#1]#2{%
1183   \def\eql@tmp{#1}%
1184   \ifx\eql@tmp\eql@testopt@default\else
1185     \eql@tags@add@ref{#1}%
1186   \fi
1187   \eql@tags@add@tag{#2}%
1188 }

```

TODO: describe

```

1189 \def\eql@tags@set@tag#1{%
1190   \ifdefined\eql@tags@warn
1191     \ifdefined\eql@tags@tag
1192       \eql@warn@tag@multiple
1193     \fi
1194   \fi
1195   \def\eql@tags@tag{#1}%
1196 }

```

TODO: describe

```

1197 \def\eql@tags@set@ref#1{%
1198   \ifdefined\eql@tags@warn
1199     \ifdefined\eql@tags@ref
1200       \eql@warn@ref@multiple
1201     \fi
1202   \fi
1203   \def\eql@tags@ref{#1}%
1204 }

```

TODO: describe

```
1205 \def\eq\@tags@add@tag#1{%
1206   \ifdefined\eq\@tags@autotag
1207     \eq\@tags@autoenable
1208   \fi
1209   \protected@edef\eq\tmp{\noexpand\eq\@tags@set@tag{#1}}%
1210   \global\eq\appendmacro\eq\@tags@container\eq\tmp
1211 }
```

TODO: describe

```
1212 \def\eq\@tags@add@ref#1{%
1213   \protected@edef\eq\tmp{\noexpand\eq\@tags@set@ref{#1}}%
1214   \global\eq\appendmacro\eq\@tags@container\eq\tmp
1215 }
```

tags@add@tagform@off

```
1216 \def\eq\@tags@add@tagform@off{%
1217   \global\eq\append\eq\@tags@container{\let\eq\@tags@tagform\@firstofone}%
1218 }
```

TODO: describe

```
1219 \def\eq\@tags@addblock@tag#1{%
1220   \protected@edef\eq\tmp{\noexpand\eq\@tags@set@tag{#1}}%
1221   \eq\appendmacro\eq\@tags@container@block\eq\tmp
1222 }
```

TODO: describe

```
1223 \def\eq\@tags@addblock@ref#1{%
1224   \protected@edef\eq\tmp{\noexpand\eq\@tags@set@ref{#1}}%
1225   \eq\appendmacro\eq\@tags@container@block\eq\tmp
1226 }
```

TODO: describe

```
1227 \def\eq\@tags@addblock@tagform@off{%
1228   \eq\append\eq\@tags@container@block{\let\eq\@tags@tagform\@firstofone}%
1229 }
```

Raise Tags.

\raisetag

```
1230 \def\eq\@raisetag@default{%
1231   \eq\warn@here\raisetag
1232   \eq\raisetag@gobble
1233 }
1234 \def\eq\@raisetag@gobble{%
1235   \eq\ampprotecttwo\eq@ifstar@tight\@gobble\@gobble
1236 }
```

TODO: describe

```
1237 \eq\amsmath@let\raisetag\eq\@raisetag@default
1238 \def\eq\@raisetag{%
1239   \eq\ampprotecttwo\eq@ifstar@tight\eq\@tags@add@raiseshift\eq\@raisetag@test
1240 }
```

```

1241 \def\eq@raisetag@test#1{%
1242   \def\eq@tmpa{#1}\def\eq@tmpb{!}%
1243   \ifx\eq@tmpa\eq@tmpb
1244     \eq@tags@add@forceraise
1245   \else
1246     \eq@tags@add@raisesmash{#1}%
1247   \fi
1248 }

1249 \def\eq@tags@add@raiseshift#1{%
1250   \global\eq@appendexpand\eq@tags@container{%
1251     \advance\eq@tagpos@shift@the\glueexpr#1\relax\relax}%
1252 }

1253 \def\eq@tags@add@raisesmash#1{%
1254   \dimen@glueexpr#1\relax
1255   \ifdim\dimen@<\z@
1256     \global\eq@appendexpand\eq@tags@container{%
1257       \advance\eq@tagpos@smashdown@the\dimen@relax}%
1258   \else
1259     \global\eq@appendexpand\eq@tags@container{%
1260       \advance\eq@tagpos@smashup@the\dimen@relax}%
1261   \fi
1262 }

1263 \def\eq@tags@add@forceraise{%
1264   \global\eq@append\eq@tags@container{\let\eq@tagpos@reserve\eq@false}%
1265 }

```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```

1266 \def\eq@numbering@settools{%
1267   \let\label\eq@label
1268   \let>tag\eq@tag
1269   \let\raisetag\eq@raisetag
1270   \let\numberhere\eq@numberhere
1271   \let\numbernext\eq@numbernext
1272 }

```

TODO: not necessary anymore

```

1273 \def\eq@numbering@settools@gobble{%
1274   \let\label\eq@label@gobble
1275   \let>tag\eq@tag@gobble
1276   \let\raisetag\eq@raisetag@gobble
1277   \let\numberhere\relax
1278   \let\numbernext\relax
1279 }

1280 \def\eq@numbering@autoblock{%
1281   \begingroup
1282     \let\eq@tags@warn\eq@false
1283     \eq@tags@container@block
1284     \ifdefined\eq@tags@autolabel

```

```

1285     \ifdefined\eql@tags@label
1286     \global\@eqnswtrue
1287     \fi
1288   \fi
1289   \ifdefined\eql@tags@autotag
1290     \ifdefined\eql@tags@tag
1291       \global\@eqnswtrue
1292     \fi
1293   \fi
1294 \endgroup
1295 }

1296 \def\eql@numbering@warnunused{%
1297   \ifdefined\eql@tags@label
1298     \eql@warn@label@unused
1299   \fi
1300   \ifdefined\eql@tags@name
1301     \eql@warn@name@unused
1302   \fi
1303   \ifdefined\eql@tags@tag
1304     \eql@warn@tag@unused
1305   \fi
1306   \ifdefined\eql@tags@erf
1307     \eql@warn@ref@unused
1308   \fi
1309 }

```

Single Line. **TODO:** describe

```

1310 \def\eql@numbering@single@init{%
1311   \let\eql@numbering@multi\eql@false
1312   \eql@numbering@settools
1313   \eql@numbering@eqnswinit
1314   \eql@numbering@autoblock
1315   \global\let\eql@tags@container\eql@tags@container@block
1316   \let\eql@tags@warn\eql@true
1317 }

1318 \def\eql@numbering@single@eval{%
1319   \ifnum\eql@tagpos@row@=\m@ne
1320     \@eqnswfalse
1321   \fi
1322 }

```

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

```

1323 \def\eql@numbering@measure@init{%
1324   \eql@numbering@settools
1325   \ifdefined\eql@numbering@multi\else
1326     \eql@numbering@eqnswinit
1327     \eql@numbering@autoblock
1328   \fi
1329   \global\let\eql@tags@container\eql@tags@container@block
1330   \let\eql@tags@warn\eql@true
1331 }

```

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


```

1332 \def\eql@numbering@measure@line@begin{%
1333   \ifdefined\eql@numbering@multi
1334     \global\eql@numbering@eqnswinit
1335   \fi
1336 }

```

TODO: describe

```

1337 \def\eql@numbering@measure@blocktag{%
1338   \ifdefined\eql@numbering@multi
1339     \eqnswfalse
1340   \else
1341     \ifnum\eql@tagpos@row@=\m@ne
1342       \eqnswfalse
1343     \fi
1344     \ifnum\eql@totalrows@=\z@
1345       \eqnswfalse
1346     \fi
1347   \fi
1348 }

```

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!

```

1349 \def\eql@numbering@print@init{%
1350   \let\eql@tags@warn\eql@false
1351   \ifdefined\eql@numbering@multi
1352     \eql@numbering@settools
1353     \global\let\eql@tags@container\eql@tags@container@block
1354   \else
1355     \let\eql@tags@container@block\eql@tags@container
1356     \eql@numbering@settools@gobble
1357   \fi
1358 }

```

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

```

1359 \def\eql@numbering@print@block@begin{%
1360   \ifdefined\eql@numbering@multi\else
1361     \ifnum\eql@tagpos@row@>\z@
1362       \eql@tags@makeblockanchor
1363       \global\eql@appendexpand\eql@tags@container@block{%
1364         \def\noexpand\eql@tags@anchor{%
1365           \unexpanded\expandafter{\eql@tags@anchor}}}%
1366     \fi
1367   \fi
1368   \ifdefined\eql@numbering@subeq@use
1369     \eql@tags@printsbeqlabel
1370   \fi
1371 }

```

TODO: describe

```

1372 \def\eql@numbering@print@line@begin{%
1373   \ifdefined\eql@numbering@multi
1374     \global\eql@numbering@eqnswinit
1375   \fi
1376 }

```

TODO: describe

```
1377 \def\eql@numbering@print@line@eval{%
1378   \ifdefined\eql@numbering@multi
1379     \if@eqnsw
1380       \eql@tags@container
1381     \fi
1382   \else
1383     \ifnum\eql@tagpos@row@=\eql@row@
1384       \@eqnswtrue
1385       \eql@tags@container@block
1386     \else
1387       \@eqnswfalse
1388     \fi
1389   \fi
1390 }
```

5.5 Positioning

TODO: describe

```
1391 \def\eql@tagpos@single@eval{%
1392   \if@eqnsw
1393     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1394     \ifnum\eql@tagpos@row@>\@ne
1395       \eql@tagpos@row@\@ne
1396     \fi
1397     \ifdefined\eql@tagpos@doconvert
1398       \let\eql@tagpos@continuous\eql@true
1399     \fi
1400     \ifdefined\eql@tagpos@continuous
1401       \eql@tagpos@single@eval@continuous
1402     \fi
1403   \else
1404     \eql@tagpos@row@\z@
1405   \fi
1406   \eql@tagpos@prevrow@\z@
1407   \eql@tagpos@headroom@\z@
1408   \eql@tagpos@footroom@\z@
1409 }
```

TODO: describe

```
1410 \def\eql@tagpos@single@eval@continuous{%
1411   \ifnum\eql@tagpos@row@>\z@
1412     \eql@tagpos@target@\eql@tagpos@shift@
1413   \else
1414     \eql@tagpos@target@\dimexpr\eql@line@height@
1415       -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1416   \fi
1417   \eql@tagpos@row@\@ne
1418   \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1419     \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1420     \eql@tagpos@target@\z@
1421   \fi
1422 }
```

TODO: describe

```

1423 \def\eql@tagpos@adjust@eval{%
1424   \if@eqnsw
1425     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1426     \ifnum\eql@tagpos@row@>\eql@totalrows@
1427       \eql@tagpos@row@\eql@totalrows@
1428     \fi
1429     \ifdefined\eql@tagpos@doconvert
1430       \let\eql@tagpos@continuous\eql@true
1431     \fi
1432     \ifdefined\eql@tagpos@continuous
1433       \ifnum\eql@tagpos@row@>\z@
1434         \eql@tagpos@adjust@eval@convert
1435       \fi
1436       \eql@tagpos@adjust@eval@continuous
1437     \fi
1438   \else
1439     \eql@tagpos@row@\z@
1440     \eql@tagpos@prevrow@\z@
1441   \fi
1442 }

```

TODO: describe

```

1443 \def\eql@tagpos@adjust@eval@convert{%
1444   \eql@tagpos@current@\z@
1445   \eql@dimensions@for{%
1446     \ifnum\eql@row@<\eql@tagpos@row@
1447       \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1448         +\eql@line@height@+\eql@line@depth@\relax
1449     \fi
1450     \ifnum\eql@row@=\eql@tagpos@row@
1451       \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1452         +\eql@line@height@-\eql@tagheight@block@\relax
1453     \fi
1454   }%
1455 }

```

TODO: describe

```

1456 \def\eql@tagpos@adjust@eval@continuous{%
1457   \dimen@\dimexpr\eql@tagpos@current@-\eql@tagpos@shift@\relax
1458   \eql@tagpos@row@\eql@totalrows@
1459   \eql@tagpos@prevrow@\z@
1460   \eql@tagpos@headroom@\z@
1461   \eql@tagpos@footroom@\z@
1462   \eql@dimensions@for{%
1463     \ifnum\eql@tagpos@row@=\eql@totalrows@
1464       \eql@tagpos@headroom@\eql@line@interline@
1465       \eql@tagpos@target@\dimexpr\eql@line@interline@
1466         +\eql@line@height@-\dimen@-\eql@tagheight@block@\relax
1467     \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1468       \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1469       \advance\dimen@\eql@tagpos@target@
1470       \eql@tagpos@target@\z@
1471     \fi
1472     \ifdim\eql@tagpos@target@>%
1473       \ifdefined\eql@tagleft-1sp\relax\else\z@\fi
1474       \eql@tagpos@row@\eql@row@
1475       \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1476     \fi

```

```

1477     \advance\dimen@-\dimexpr\eq@line@interline@
1478         +\eq@line@depth@+\eq@line@height@\relax
1479     \fi
1480     \ifnum\eq@row@=\numexpr\eq@tagpos@row@+\@ne\relax
1481         \eq@tagpos@footroom@\eq@line@interline@
1482     \fi
1483 }%
1484 }

```

TODO: describe

```

1485 \def\eq@tagpos@print@line@eval{%
1486     \ifdefined\eq@tagpos@continuous
1487         \eq@tagpos@print@line@eval@continuous
1488     \else
1489         \eq@tagpos@print@line@eval@discrete
1490     \fi
1491 }

```

TODO: describe

```

1492 \def\eq@tagpos@print@line@eval@continuous{%
1493     \if@eqnsw
1494         \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@smashup@\relax
1495         \dp\eq@tagbox@\dimexpr\dp\eq@tagbox@-\eq@tagpos@smashdown@\relax
1496         \eq@tagpos@plain@\eq@tagpos@target@
1497         \@tempdima\dimexpr\eq@line@height@+\eq@tagpos@headroom@
1498             -\ht\eq@tagbox@\relax
1499         \@tempdimb\dimexpr-\eq@line@depth@-\eq@tagpos@footroom@
1500             +\dp\eq@tagbox@\relax
1501         \ifnum\eq@row@=\@ne
1502             \@tempdima.5\maxdimen
1503         \fi
1504         \ifnum\eq@row@=\eq@totalrows@
1505             \@tempdimb-.5\maxdimen
1506         \fi
1507         \ifdim\eq@tagpos@plain@>\@tempdima
1508             \ifdim\eq@tagpos@plain@>\@tempdimb
1509                 \ifdim\@tempdima>\@tempdimb
1510                     \eq@tagpos@plain@\@tempdima
1511                 \else
1512                     \eq@tagpos@plain@\@tempdimb
1513                 \fi
1514             \fi
1515         \else
1516             \ifdim\eq@tagpos@plain@<\@tempdimb
1517                 \ifdim\@tempdima>\@tempdimb
1518                     \eq@tagpos@plain@\@tempdimb
1519                 \else
1520                     \eq@tagpos@plain@\@tempdima
1521                 \fi
1522             \fi
1523         \fi
1524         \ifnum\eq@tagpos@prevrow@>\z@
1525             \eq@tagpos@raised@\dimexpr\eq@line@height@+\dp\eq@tagbox@\relax
1526             \ifdim\eq@tagpos@raised@>\eq@tagpos@plain@\else
1527                 \eq@tagpos@raised@\eq@tagpos@plain@
1528                 \let\eq@tagpos@reserve\eq@false
1529             \fi
1530         \else

```

```

1531 \ifdim\eql@tagpos@target@>%
1532 \ifdefined\eql@tagsleft-1sp\relax\else\z@ \fi
1533 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1534 \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1535 \eql@tagpos@raised@\eql@tagpos@plain@
1536 \let\eql@tagpos@reserve\eql@false
1537 \fi
1538 \else
1539 \eql@tagpos@raised@\dimexpr-\eql@line@depth@
1540 -\ht\eql@tagbox@\relax
1541 \ifdim\eql@tagpos@raised@<\eql@tagpos@plain@\else
1542 \eql@tagpos@raised@\eql@tagpos@plain@
1543 \let\eql@tagpos@reserve\eql@false
1544 \fi
1545 \fi
1546 \fi
1547 \else
1548 \ifnum\eql@tagpos@prevrow@=\eql@row@
1549 \eql@tagwidth@\eql@tagwidth@block@
1550 \else
1551 \let\eql@tagpos@reserve\eql@false
1552 \fi
1553 \fi
1554 }

```

TODO: describe

```

1555 \def\eql@tagpos@print@line@eval@discrete{%
1556 \if@eqnsw
1557 \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1558 \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1559 \eql@tagpos@plain@\eql@tagpos@shift@
1560 \eql@tagpos@headroom@\z@
1561 \eql@tagpos@footroom@\z@
1562 \ifdim\eql@tagpos@shift@>%
1563 \ifdefined\eql@tagsleft-1sp\relax\else\z@ \fi
1564 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1565 \else
1566 \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1567 \fi
1568 \else
1569 \let\eql@tagpos@reserve\eql@false
1570 \fi
1571 }

```

TODO: describe

```

1572 \def\eql@tagpos@print@line@end{%
1573 \ifdefined\eql@tagpos@continuous
1574 \ifnum\eql@tagpos@prevrow@=\eql@row@
1575 \ifdefined\eql@tagpos@reserve
1576 \global\eql@appendexpand\eql@tags@container@block{%
1577 \advance\eql@tagpos@headroom@the\dimexpr\eql@line@height@
1578 +\eql@line@depth@\relax\relax}%
1579 \eql@displaybreak@star\@M
1580 \fi
1581 \fi
1582 \fi
1583 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1584 \let\eqL@SK@loaded\eqL@false
1585 \let\eqL@SK@label\@gobble
1586 \let\eqL@SK@clearlabel\@empty
1587 \let\eqL@SK@setlabel\@gobble
1588 \let\eqL@SK@printlabel@right\@empty
1589 \let\eqL@SK@printlabel@left\@empty
1590 \let\eqL@SK@printlabel@line\@empty
1591 \def\eqL@label@clean{\eqL@label@org}
1592 \AddToHook{package/showkeys/after}{%
1593   \let\eqL@SK@loaded\eqL@true
1594   \def\eqL@SK@label#1{\SK@\SK@@label#1}%
1595   \def\eqL@SK@clearlabel{\let\eqL@SK@lab\relax}%
1596   \eqL@SK@clearlabel
1597   \def\eqL@SK@@label#1>#2\SK@{%
1598     \def\eqL@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1599   }%
1600   \def\eqL@SK@setlabel#1{\SK@\eqL@SK@@label#1}%
1601   \def\eqL@SK@printlabel@right{%
1602     \ifx\eqL@SK@lab\relax\else
1603       \rlap{\kern\marginparsep\eqL@SK@lab}%
1604       \eqL@SK@clearlabel
1605     \fi
1606   }%
1607   \def\eqL@SK@printlabel@left{%
1608     \ifx\eqL@SK@lab\relax\else
1609       \llap{\eqL@SK@lab\kern\marginparsep}%
1610       \eqL@SK@clearlabel
1611     \fi
1612   }%
1613   \def\eqL@SK@printlabel@line{%
1614     \ifx\eqL@SK@lab\relax\else
1615       \dimen@\prevdepth
1616       \nointerlineskip
1617       \ifdefined\eqL@tagsleft
1618         \llap{%
1619           \eqL@SK@lab
1620           \kern\marginparsep
1621         }%
1622       \eqL@SK@clearlabel
1623     \else
1624       \rlap{%
1625         \dimen@\displaywidth
1626         \advance\dimen@\marginparsep
1627         \kern\dimen@
1628         \eqL@SK@lab
1629       }%
1630     \fi
1631     \eqL@SK@clearlabel
1632     \prevdepth\dimen@
1633   \fi
1634 }%
1635 \let\eqL@label@org\label
1636 \def\eqL@label@clean{\let\SK@\@gobbletwo\eqL@label@org}
1637 }

```

Labels.

`eql@composetag@label` **TODO:** describe

```
1638 \def\eql@composetag@label{%
1639   \eql@SK@clearlabel
1640   \ifdefined\eql@tags@label
1641     \eql@SK@setlabel\eql@tags@label
1642     \ifdefined\eql@tags@name
1643       \let\@currentlabelname\eql@tags@name
1644     \else
1645       \let\@currentlabelname\eql@tags@name@generic
1646     \fi
1647     \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1648   \fi
1649 }
```

TODO: describe

```
1650 \def\eql@tags@printsubeqlabel{%
1651   \eql@tags@container@parent
1652   \ifdefined\eql@tags@label
1653     \eql@tags@makeblockanchor
1654     \eql@SK@setlabel\eql@tags@label
1655     \begingroup
1656       \def\@currentcounter{equation}%
1657       \eql@tags@anchor
1658       \let\@currentlabelname\eql@tags@name@generic
1659       \protected@edef\@currentlabel{\p@equation\theparentequation}%
1660       \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1661     \endgroup
1662     \eql@SK@printlabel@line
1663   \fi
1664 }
```

Hyperref Anchors. **TODO:** describe

```
1665 \let\eql@Hy@anchor\@gobble
1666 \AddToHook{package/hyperref/after}{%
1667   \def\eql@Hy@anchor#1{%
1668     \Hy@raisedlink{\hyper@anchor{#1}}%
1669   }%
1670 }
```

TODO: describe

```
1671 \def\eql@tags@makeblockanchor{%
1672   \eql@tags@glabel@step
1673   \eql@Hy@anchor\eql@tags@glabel
1674   \edef\eql@tags@anchor{%
1675     \def\noexpand\thepage{\thepage}%
1676     \def\noexpand\@currentHref{\eql@tags@glabel}%
1677   }%
1678 }
```

TODO: describe

`ql@composetag@anchor`

```
1679 \def\eql@composetag@anchor{%
```

```

1680 \ifdefined\eql@tags@tag
1681   \def\@currentcounter{equation}%
1682   \ifdefined\eql@tags@ref
1683     \let\@currentlabel\eql@tags@ref
1684   \else
1685     \protected@edef\@currentlabel{\p@equation\eql@tags@tag}%
1686   \fi
1687   \eql@tags@glabel@step
1688   \edef\@currentHref{\eql@tags@glabel}%
1689   \eql@Hy@anchor\@currentHref
1690 \else
1691   \refstepcounter{equation}%
1692   \protected@edef\eql@tags@tag{\theequation}%
1693 \fi
1694 \eql@tags@anchor
1695 }

```

Tag Layout. **TODO:** describe

```

1696 \def\eql@tags@taglayout@set@direct#1{%
1697   \def\eql@tags@taglayout##1{#1}%
1698 }
1699 \def\eql@tags@taglayout@set#1{%
1700   \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1701 }

```

TODO: describe

```

1702 \def\eql@tags@tagform@set@direct#1{%
1703   \def\eql@tags@tagform##1{#1}%
1704 }
1705 \def\eql@tags@tagform@set#1#2#3{%
1706   \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1707 }

1708 \eql@tags@taglayout@set{#1}
1709 \eql@tags@tagform@set({#1})
1710 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1711 \protected\def\tagform{\eql@tags@tagform}
1712 \protected\def\tagbox{\eql@tags@taglayout}
1713 \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:

```

1714 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1715 \def\eql@composetag@tag{%
1716   \eql@tagging@tagbegin
1717   \eql@tags@frame@cmd{%
1718     \eql@tags@taglayout{%
1719       \eql@tags@tagform\eql@tags@tag
1720       \eql@tagging@tagsave
1721     }%
1722   }%
1723   \eql@tagging@tagend
1724 }

```


5.7 Tag Composition

TODO: describe

```
1725 \def\eql@composetag@measure{%
1726   \ifdefined\eql@tags@tag\else
1727     \stepcounter{equation}%
1728     \let\eql@tags@tag\theequation
1729   \fi
1730   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1731   \ifdefined\eql@numbering@multi
1732     \global\let\eql@tags@container\eql@tags@container@clear
1733   \fi
1734 }
```

TODO: describe

```
1735 \def\eql@composetag@print{%
1736   \eql@composetag@anchor
1737   \eql@composetag@label
1738   \ifdefined\eql@tags@left
1739     \eql@SK@printlabel@left
1740     \eql@composetag@tag
1741   \else
1742     \eql@composetag@tag
1743     \eql@SK@printlabel@right
1744   \fi
1745   \global\let\eql@tags@container\eql@tags@container@clear
1746 }
```

TODO: describe

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

```
1747 \def\eql@tagbox@make#1{%
1748   \setbox\eql@tagbox\hbox{\eql@strut@tag\@lign#1}%
1749   \eql@tagwidth@wd\eql@tagbox@
1750   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1751     \eql@tagwidth@\eql@tagwidthmin@
1752   \fi
1753   \advance\eql@tagwidth@\eql@tagsepmin@
1754 }
```

TODO: describe

```
1755 \def\eql@tagbox@print@adjustheadroom{%
1756   \dimen@dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1757   \ifdim\dimen@>\z@
1758     \ifdim\dimen@>\eql@tagpos@headroom@
1759       \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1760     \else
1761       \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1762     \fi
1763   \fi
1764 }
```

TODO: describe

```
1765 \def\eql@tagbox@print@adjustfootroom{%
1766   \dimen@dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1767   \ifdim\dimen@>\z@
1768     \ifdim\dimen@>\eql@tagpos@footroom@
```

```

1769     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1770   \else
1771     \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1772   \fi
1773 \fi
1774 }

```

TODO: describe

```

1775 \def\eql@tagbox@print@extendabove{%
1776   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1777   \ifdim\dimen@>\z@
1778     \global\eql@appendexpand\eql@display@container{%
1779       \eql@display@aboveextend@the\dimen@\relax}%
1780   \fi
1781 }

```

TODO: describe

```

1782 \def\eql@tagbox@print@extendbelow{%
1783   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1784   \ifdim\dimen@>\z@
1785     \global\eql@appendexpand\eql@display@container{%
1786       \eql@display@belowextend@the\dimexpr\dimen@\relax}%
1787   \fi
1788 }

```

TODO: describe

```

1789 \def\eql@tagbox@print@prepare{%
1790   \ifdefined\eql@tagpos@reserve
1791     \eql@tagpos@current@\eql@tagpos@plain@
1792   \else
1793     \eql@tagpos@current@\eql@tagpos@raised@
1794   \fi
1795   \ifdim\eql@tagpos@headroom@>\z@
1796     \eql@tagbox@print@adjusttheadroom
1797   \fi
1798   \ifdim\eql@tagpos@footroom@>\z@
1799     \eql@tagbox@print@adjustfootroom
1800   \fi
1801   \ifnum\eql@row@=\@ne
1802     \eql@tagbox@print@extendabove
1803   \fi
1804   \ifnum\eql@row@=\eql@totalrows@
1805     \eql@tagbox@print@extendbelow
1806   \fi
1807 }

```

TODO: describe

```

1808 \def\eql@tagbox@print@tagsright{%
1809   \eql@tagbox@print@prepare
1810   \kern-\wd\eql@tagbox@
1811   \raise\eql@tagpos@current@\box\eql@tagbox@
1812 }

```

TODO: describe

```

1813 \def\eql@tagbox@print@tagsleft{%
1814   \eql@display@firstavail@set\z@
1815   \eql@tagbox@print@prepare

```

```

1816 \wd\eql@tagbox@\z@
1817 \raise\eql@tagpos@current@\box\eql@tagbox@
1818 }

```

ql@tagbox@print@cell

```

1819 \def\eql@tagbox@print@cell{%
1820 \eql@tagging@tagaddbox
1821 \ifdefined\eql@tagsleft
1822 \ifdefined\eql@tagpos@reserve
1823 \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1824 \let\eql@tagpos@reserve\eql@false
1825 \fi
1826 \fi
1827 \if@eqnsw
1828 \eql@tagbox@print@tagsleft
1829 \fi
1830 \kern\displaywidth
1831 \else
1832 \kern\displaywidth
1833 \ifdefined\eql@tagpos@reserve
1834 \ifdim\eql@tagwidth@>%
1835 \dimexpr\displaywidth-\eql@line@width@+\eql@tagfuzz@\relax
1836 \let\eql@tagpos@reserve\eql@false
1837 \fi
1838 \fi
1839 \if@eqnsw
1840 \eql@tagbox@print@tagsright
1841 \fi
1842 \fi
1843 }

```

6 Subequation Numbering

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

6.1 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:

```

1844 \eql@amsmath@undefine\c@parentequation
1845 \eql@amsmath@undefine\theparentequation
1846 \ifdefined\c@parentequation\else
1847 \newcounter{parentequation}
1848 \fi

```

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

```

1849 \def\eql@subequations@template{\theparentequation\alph{equation}}
1850 \def\eql@subequations@template@set#1{\def\eql@tmp##1##2{#1}%
1851 \edef\eql@subequations@template{%
1852 \unexpanded\expandafter{\eql@tmp\theparentequation{equation}}}}

```

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

```
1853 \let\eq\@subequations@active\eq\false
```

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

```
1854 \def\eq\@subequations@init{%
1855   \edef\eq\@subequations@restorecounter{%
1856     \global\c@equation\the\c@equation\relax}%
1857   \eq\@tags@container@block
1858   \ifdefined\eq\@tags@tag
1859     \eq\@tags@glabel@step
1860     \protected@edef\theHparentequation{\eq\@tags@glabel}%
1861     \protected@edef\theparentequation{\eq\@tags@tag}%
1862   \else
1863     \advance\c@equation\@ne
1864     \protected@edef\theparentequation{\theequation}%
1865     \ifdefined\theHequation
1866       \protected@edef\theHparentequation{\theHequation}%
1867     \fi
1868   \fi
1869   \global\c@parentequation\c@equation
1870   \global\c@equation\z@
1871   \let\theequation\eq\@subequations@template
1872   \def\theHequation{\theHparentequation.\arabic{equation}}}%
1873 }
```

`eq\@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:

```
1874 \def\eq\@subequations@close{%
1875   \ifnum\c@equation=\z@
1876     \eq\@subequations@restorecounter
1877   \else
1878     \global\c@equation\c@parentequation
1879   \fi
1880 }
```

6.2 Environment

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

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

```
1881 \def\eq\@subequations@start{%
1882   \let\eq\@tags@container@block\eq\@tags@container@clear
1883   \eq\@nextopt@process{subequations}%
1884   \eq\@subequations@init
1885   \eq\@tags@glabel@step
```

```

1886 \edef\eqL@subequations@currentHref{\eqL@tags@glabel}%
1887 \eqL@Hy@anchor\eqL@subequations@currentHref
1888 \edef\eqL@subequations@thepage{\thepage}%
1889 \def\@currentcounter{equation}%
1890 \let\@currentHref\eqL@subequations@currentHref
1891 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1892 \eqL@tags@container@block
1893 \ifdefined\eqL@tags@name
1894   \let\@currentlabelname\eqL@tags@name
1895 \else
1896   \let\@currentlabelname\eqL@tags@name@generic
1897 \fi
1898 \let\eqL@subequations@active\eqL@true
1899 \ifdefined\eqL@tags@label
1900   \eqL@SK@label\eqL@tags@label
1901 \fi
1902 \ignorespaces
1903 }

```

`eqL@subequations@end` End the subequations environment. Issue the label if one has been specified and an equation number has actually been used. Then close subequations mode: **TODO**: how about tag* ?! also within equations!

```

1904 \def\eqL@subequations@end{%
1905   \ifnum\c@equation>\z@
1906     \eqL@tags@container@block
1907     \ifdefined\eqL@tags@label
1908       \begingroup
1909         \def\@currentcounter{equation}%
1910         \let\thepage\eqL@subequations@thepage
1911         \let\@currentHref\eqL@subequations@currentHref
1912         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1913         \ifdefined\eqL@tags@name
1914           \let\@currentlabelname\eqL@tags@name
1915         \else
1916           \let\@currentlabelname\eqL@tags@name@generic
1917         \fi
1918         \expandafter\eqL@label@clean\expandafter{\eqL@tags@label}%
1919       \endgroup
1920     \fi
1921   \fi
1922   \eqL@subequations@close
1923 }

```

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

```

1924 \newenvironment{eqL@subequations}{%
1925   \eqL@verbose@info\eqL@verbose@msg@enterenv
1926   \eqL@subequations@testall\eqL@subequations@start%
1927 }{%
1928   \eqL@subequations@end
1929   \ignorespacesafterend
1930   \eqL@verbose@info\eqL@verbose@msg@leaveenv
1931 }

```

TODO: describe

```

1932 \def\eqL@subequations@testall{\eqL@parseopt@env\eqL@subequations@testall@parse}

```

```

1933 \def\eq@subequations@testall@parse{%
1934   \ifx\eq@parseopt@token[%]
1935     \let\eq@parseopt@next\eq@parseopt@opt
1936   \fi
1937   \ifx\eq@parseopt@token\eq@atxi
1938     \let\eq@parseopt@next\eq@parseopt@label
1939   \fi
1940   \ifx\eq@parseopt@token\eq@atxii
1941     \let\eq@parseopt@next\eq@parseopt@label
1942   \fi
1943   \ifx\eq@parseopt@token\label
1944     \let\eq@parseopt@next\eq@parseopt@end
1945   \fi
1946 }

```

6.3 Subequation Scheme

TODO: describe

```

1947 \def\eq@numbering@subeq@init{%
1948   \let\eq@save@theequation\theequation
1949   \let\eq@save@theHequation\theHequation
1950   \eq@subequations@init
1951   \let\eq@tags@container@parent\eq@tags@container@block
1952   \let\eq@tags@container@block\eq@tags@container@clear
1953 }

```

TODO: describe

```

1954 \def\eq@numbering@subeq@test{%
1955   \ifnum\eq@tagrows@<\tw@
1956     \let\eq@tags@container@block\eq@tags@container@parent
1957     \let\eq@numbering@subeq@use\eq@false
1958     \let\theequation\eq@save@theequation
1959     \let\theHequation\eq@save@theHequation
1960     \eq@subequations@restorecounter
1961   \fi
1962 }

```

TODO: describe

```

1963 % \TODO note must not use setcounter here (when calc is loaded)
1964 \def\eq@numbering@subeq@close{%
1965   \eq@subequations@close
1966 }

```

7 Display Equations Support

TODO: describe

```

1967 \let\eq@display@injectbefore\@undefined
1968 \let\eq@display@injectafter\@undefined
1969 \let\eq@interline@container\@undefined
1970 \def\eq@interline@container@clear{%
1971   \eq@displaybreak@open@\@MM
1972   \eq@vspaceskip@\z@skip
1973 }

```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1974 \interdisplaylinepenalty\@M
```

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

```
1975 \def\eqldgetdsp@pen#1{%
1976   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1977 }
```

TODO: allow a displaybreak before equations

```
1978 \protected\def\eqldisplaybreak@default{%
1979   \eqldwarning{Invalid use of \string\displaybreak}{}%
1980   \eqldteststaropt@loose\@gobble\eqldgobbleopt{}}
1981 \eqldamsmath@after{\let\eqldisplaybreak@default\displaybreak}
1982 \eqldamsmath@let\displaybreak\eqldisplaybreak@default

1983 \newcount\eqldisplaybreak@pen@
1984 \newcount\eqldisplaybreak@prepen@
1985 \newcount\eqldisplaybreak@postpen@
```

TODO: describe

```
1986 \protected\def\eqldisplaybreak{%
1987   \relax
1988   \eqldampprocttwo\eqldteststaropt@tight
1989   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1990 }

1991 \def\eqldisplaybreak@star#1{%
1992   \global\eqldappendexpand\eqldinterline@container{%
1993     \eqldisplaybreak@pen@\the\numexpr#1\relax\relax}%
1994 }

1995 \def\eqldisplaybreak@level[#1]{%
1996   \ifnum#1<\z@
1997     \global\eqldappend\eqldinterline@container{\eqldisplaybreak@pen@\@MM}%
1998   \else
1999     \global\eqldappendexpand\eqldinterline@container{%
2000       \eqldisplaybreak@pen@-\@getpen{#1}\relax}%
2001   \fi
2002 }
```

TODO: describe

```
2003 \def\eqldisplaybreak@pre#1{%
2004   \ifnum#1<\z@
2005     \eqldisplaybreak@prepen@\@MM
2006   \else
2007     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
2008   \fi
2009 }
```

TODO: describe

```
2010 \def\eqldisplaybreak@post#1{%
```

```

2011 \ifnum#1<\z@
2012   \eqldisplaybreak@postpen@MM
2013 \else
2014   \eqldisplaybreak@postpen@-\getpen{#1}\relax
2015 \fi
2016 }

```

TODO: describe

```

2017 \def\eqldisplaybreak@inter#1{%
2018   \ifnum#1<\z@
2019     \interdisplaylinepenaltyM
2020 \else
2021   \interdisplaylinepenalty\eqldspopen{#1}\relax
2022 \fi
2023 }

```

7.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```

2024 \newskip\eqlvspaceskip@
2025 \let\eqlvspace@org\vspace
2026 \def\eqlvspace{%
2027   \ifvmode
2028     \expandafter\eqlvspace@immediate
2029   \else
2030     \expandafter\eqlvspace@line
2031   \fi
2032 }

```

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

```

2033 \def\eqlvspace@line{%
2034   \eq@ifstar@loose\eqlvspace@addfixedbefore\eqlvspace@add
2035 }
2036 \def\eqlvspace@add#1{%
2037   \global\eql@appendexpand\eql@interline@container{%
2038     \advance\eqlvspaceskip@\the\glueexpr#1\relax\relax}}
2039 \def\eqlvspace@addfixedbefore#1{%
2040   \global\eql@appendexpand\eql@interline@container{%
2041     \noexpand\eql@append\noexpand\eqldisplay@injectbefore{%
2042       \skip@\the\glueexpr#1\relax\relax
2043       \penaltyM
2044       \vskip\skip@
2045       \global\advance\eql@line@interline@\skip@
2046     }%
2047   }%
2048 }
2049 \def\eqlvspace@addfixedafter#1{%
2050   \global\eql@appendexpand\eql@interline@container{%
2051     \noexpand\eql@append\noexpand\eqldisplay@injectafter{%
2052       \dimen@\prevdepth
2053       \hrule\@height\z@
2054       \skip@\the\glueexpr#1\relax\relax

```



```

2055     \penalty\@M
2056     \vskip\skip@
2057     \global\advance\eql@line@interline@\skip@
2058     \prevdepth\dimen@
2059 }%
2060 }%
2061 }

```

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

```

2062 \def\eql@vspace@immediate{%
2063   \noalign\bgroup
2064     \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
2065 }
2066 \def\eql@vspace@fixed#1{%
2067   \skip@\glueexpr#1\relax
2068   \ifnum\eql@row@=\@ne
2069     \global\eql@appendexpand\eql@display@container{%
2070       \advance\eql@abovespace@\the\skip@\relax}%
2071   \else\ifnum\eql@row@>\eql@totalrows@
2072     \global\eql@appendexpand\eql@display@container{%
2073       \advance\eql@belowspace@\the\skip@\relax}%
2074   \else
2075     \dimen@\prevdepth
2076     \hrule\@height\z@
2077     \penalty\@M
2078     \vskip\skip@
2079     \global\advance\eql@line@interline@\skip@
2080     \prevdepth\dimen@
2081   \fi\fi
2082 \egroup
2083 }
2084 \def\eql@vspace@discardable#1{%
2085   \skip@\glueexpr#1\relax
2086   \ifnum\eql@row@=\@ne
2087     \global\eql@appendexpand\eql@display@container{%
2088       \advance\eql@abovespace@\the\skip@\relax}%
2089   \else\ifnum\eql@row@>\eql@totalrows@
2090     \global\eql@appendexpand\eql@display@container{%
2091       \advance\eql@belowspace@\the\skip@\relax}%
2092   \else
2093     \vskip\skip@
2094     \global\advance\eql@line@interline@\skip@
2095   \fi\fi
2096 \egroup
2097 }

```

7.3 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.

```

2098 \newbox\eql@strutbox@
2099 \def\eql@strut@depth{.3}
2100 \def\eql@strut{\copy\eql@strutbox@}
2101 \let\eql@strut@cell\eql@strut

```

```

2102 \let\eql@strut@tag\eql@strut
2103 \def\eql@strut@make{%
2104   \setbox\eql@strutbox@\hbox{%
2105     \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
2106     \eql@strut@depth\normalbaselineskip-.5\normallineskiplimit\relax
2107     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
2108   }
2109 }
2110 }
2111 }
2112 \AtBeginDocument{\eql@strut@make}

```

TODO: describe **TODO:** uses `amsmath \spread@equation`

```

2113 \def\eql@spread@set{%
2114   \ifdefined\eql@spread@reset
2115     \lineskip\normalbaselineskip
2116     \lineskiplimit\normallineskiplimit
2117     \baselineskip\normalbaselineskip
2118   \fi
2119   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
2120   +\normalbaselineskip-\baselineskip\relax
2121   \ifdim\eql@spread@>\z@
2122     \openup\eql@spread@
2123     \ifdefined\spread@equation
2124       \let\spread@equation\@empty
2125     \fi
2126   \fi
2127 }

```

7.4 Entry and Exit

TODO: describe

```

2128 \let\eql@beamerbasecolor@fix\@empty
2129 \AddToHook{package/beamerbasecolor/after}{%
2130   \def\eql@beamerbasecolor@fix{%
2131     \donotcolorouterdisplaymaths
2132     \donotcoloroutermaths
2133     \beamer@setdisplaymathcolor
2134   }%
2135 }

```

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

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

```

2136 \newskip\eql@abovespace@
2137 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

2138 \def\eql@display@enter{%
2139   \if@noskipsec\leavevmode\par\fi
2140   \ifvmode
2141     \eql@prevdepth@\prevdepth
2142     \nointerlineskip
2143     \noindent
2144   \else
2145     \eql@prevdepth@\maxdimen

```

```

2146 \fi
2147 \eql@beamerbasecolor@fix
2148 }

```

\eql@display@adjust

```

2149 \def\eql@display@adjust{%
2150   \ifdefined\eql@display@linewidth
2151     \displaywidth\glueexpr\eql@display@linewidth\relax
2152     \advance\displaywidth-\displayindent
2153   \fi
2154   \ifdefined\eql@display@marginleft
2155     \advance\displaywidth\displayindent
2156     \displayindent\glueexpr\eql@display@marginleft\relax
2157     \advance\displaywidth-\displayindent
2158   \fi
2159   \ifdefined\eql@display@marginright
2160     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
2161   \fi
2162   \ifdim\displaywidth<\z@
2163     \displaywidth\z@
2164   \fi
2165 }

```

\eql@display@init

```

2166 \def\eql@display@init{%
2167   \let\eql@display@restore\eql@display@restore@active
2168   \let\displaybreak\eql@displaybreak
2169   \let\eql@vspace@org\vspace
2170   \let\vspace\eql@vspace
2171   \let\eqncontrol\eql@control
2172   \let\eql@display@injectbefore\@empty
2173   \let\eql@display@injectafter\@empty
2174   \let\eqnpunct\eql@punct@setnext
2175   \eql@spread@set
2176   \eql@strut@make
2177   \let\eql@frame@cmd\@undefined
2178 }

```

\eql@display@print

```

2179 \def\eql@display@print{%
2180   \eql@punct@top@set
2181   \let\eql@display@container\@empty
2182   \eql@display@firstavail\z@
2183   \eql@display@aboveextend\z@
2184   \eql@display@belowextend\z@
2185   \global\let\eql@interline@container\eql@interline@container@clear
2186 }

```

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

```

2187 \def\eql@display@halign@init#1{%
2188   \eql@row@\z@
2189   \eql@prevgraf@\prevgraf
2190   \everycr{\noalign{%
2191     \global\advance\eql@row@\@ne
2192     \prevgraf\numexpr\prevgraf+\@ne\relax

```

```

2193     #1%
2194   }}%
2195 }

```

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

```

2196 \def\eqldisplay@halign@start{%
2197   \prevgraf\numexpr\eqldisplay@prevgraf@+\@ne\relax
2198   \ifdim\eqldisplay@prevdepth@=\maxdimen\else
2199     \prevdepth\eqldisplay@prevdepth@
2200   \fi
2201   \ifdim\prevdepth=-\@m\p@ \else
2202     \ifdefined\eqldisplay@height
2203       \skip@ \baselineskip
2204       \advance\skip@-\glueexpr\eqldisplay@height\relax
2205       \advance\skip@-\prevdepth\relax
2206       \ifdim\skip@<\lineskiplimit
2207         \skip@ \lineskip
2208       \fi
2209       \advance\skip@-\eqldisplay@spread@\relax
2210       \vskip\skip@
2211       \nointerlineskip
2212     \else
2213       \vskip-\eqldisplay@spread@\relax
2214     \fi
2215   \fi
2216 }

```

TODO: describe

```

2217 \def\eqldisplay@vspace{%
2218   \advance\abovedisplayskip\eqldisplay@abovespace@
2219   \advance\belowdisplayskip\eqldisplay@belowspace@
2220 }

```

TODO: describe

```

2221 \def\eqldisplay@vspace@native{%
2222   \advance\abovedisplayskip\eqldisplay@abovespace@
2223   \advance\belowdisplayskip\eqldisplay@belowspace@
2224   \advance\abovedisplayshortskip\eqldisplay@abovespace@
2225   \advance\belowdisplayshortskip\eqldisplay@belowspace@
2226 }

```

TODO: describe

```

2227 \def\eqldisplay@penalty{%
2228   \ifnum\eqldisplay@break@postpen@=\@MM\else
2229     \postdisplaypenalty\eqldisplay@break@postpen@
2230   \fi
2231   \ifnum\eqldisplay@break@open@=\@MM\else
2232     \postdisplaypenalty\eqldisplay@break@open@
2233   \fi
2234   \ifnum\eqldisplay@break@prepen@=\@MM\else
2235     \predisplayskip\eqldisplay@break@prepen@
2236   \fi
2237 }

```

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

```

2238 \def\eqldisplay@halign@end{%
2239   \eqldisplay@interline@container

```

```

2240 \eqldisplay@injectbefore
2241 \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
2242 \ifdefined\eqldisplay@depth
2243 \prevdepth\glueexpr\eqldisplay@depth\relax
2244 \fi
2245 }

```

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

```

2246 \def\eqldisplay@close{%
2247 \eqldisplay@container
2248 \ifdim\eqldisplay@firstavail@<\z@
2249 \eqldisplay@firstavail@\z@
2250 \fi
2251 \eql@skip@mode@leave@\z@
2252 \ifdim\eql@prevdepth@=\maxdimen
2253 \ifdim\predisplaysize=-\maxdimen
2254 \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
2255 \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
2256 \else
2257 \eql@skip@mode@above@\z@
2258 \eql@skip@mode@below@\z@
2259 \advance\eqldisplay@firstavail@\displayindent
2260 \ifdim\eqldisplay@firstavail@>\predisplaysize
2261 \ifcase\eql@skip@mode@short\relax
2262 \or
2263 \eql@skip@mode@above@\@ne
2264 \or
2265 \eql@skip@mode@above@\@ne
2266 \ifnum\eql@totalrows@=\@ne
2267 \eql@skip@mode@below@\@ne
2268 \fi
2269 \or
2270 \eql@skip@mode@above@\@ne
2271 \eql@skip@mode@below@\@ne
2272 \fi
2273 \fi
2274 \fi
2275 \else
2276 \ifdim\eql@prevdepth@=-\@m\p@
2277 \eql@skip@mode@above@\eql@skip@mode@top@above\relax
2278 \eql@skip@mode@below@\eql@skip@mode@top@below\relax
2279 \else
2280 \eql@skip@mode@above@\eql@skip@mode@par@above\relax
2281 \eql@skip@mode@below@\eql@skip@mode@par@below\relax
2282 \fi
2283 \fi
2284 \ifcase\eql@skip@mode@above@
2285 \or\or\or
2286 \predisplaysizepenalty\z@
2287 \or
2288 \predisplaysizepenalty\z@
2289 \fi
2290 \ifcase\eql@skip@mode@below@
2291 \or\or\or
2292 \eql@skip@mode@leave@\@ne
2293 \or

```

```

2294 \eq@skip@mode@leave@tw@
2295 \fi
2296 \ifdefined\eq@skip@force@above
2297 \eq@skip@mode@above@eq@skip@force@above\relax
2298 \fi
2299 \ifdefined\eq@skip@force@below
2300 \eq@skip@mode@below@eq@skip@force@below\relax
2301 \fi
2302 \ifdefined\eq@skip@force@leave
2303 \eq@skip@mode@leave@eq@skip@force@leave\relax
2304 \fi
2305 \ifnum\eq@skip@mode@leave@>\z@
2306 \postdisplaypenalty\z@
2307 \fi
2308 \ifcase\eq@skip@mode@above@
2309 \abovedisplayskip\glueexpr\eq@skip@long@above\relax
2310 \or
2311 \abovedisplayskip\glueexpr\eq@skip@short@above\relax
2312 \or
2313 \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
2314 \or
2315 \abovedisplayskip\glueexpr\eq@skip@par@above\relax
2316 \or
2317 \abovedisplayskip\glueexpr\eq@skip@top@above\relax
2318 \or
2319 \abovedisplayskip\z@skip
2320 \or
2321 \abovedisplayskip\glueexpr\eq@skip@med@above\relax
2322 \or
2323 \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
2324 \fi
2325 \ifcase\eq@skip@mode@below@
2326 \belowdisplayskip\glueexpr\eq@skip@long@below\relax
2327 \or
2328 \belowdisplayskip\glueexpr\eq@skip@short@below\relax
2329 \or
2330 \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
2331 \or
2332 \belowdisplayskip\glueexpr\eq@skip@par@below\relax
2333 \or
2334 \belowdisplayskip\glueexpr\eq@skip@top@below\relax
2335 \or
2336 \belowdisplayskip\z@skip
2337 \or
2338 \belowdisplayskip\glueexpr\eq@skip@med@below\relax
2339 \or
2340 \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
2341 \fi
2342 \global\eq@skip@mode@leave@eq@skip@mode@leave@
2343 \eq@interline@container
2344 \advance\eq@belowspace@\eq@vspaceskip@
2345 \eq@display@penalty
2346 \eq@display@vspace
2347 \skip@\glueexpr\eq@skip@tag@above\relax
2348 \ifdim\skip@>\abovedisplayskip
2349 \skip@\abovedisplayskip
2350 \fi
2351 \advance\abovedisplayskip-\eq@display@aboveextend@\relax

```

```

2352 \ifdim\abovedisplayskip<\skip@
2353   \abovedisplayskip\skip@
2354 \fi
2355 \skip@\glueexpr\eq\skip@tag@below\relax
2356 \ifdim\skip@>\belowdisplayskip
2357   \skip@\belowdisplayskip
2358 \fi
2359 \ifdim\eq\display@belowextend@>\z@
2360   \advance\belowdisplayskip-\eq\display@belowextend@\relax
2361   \ifdim\belowdisplayskip<\skip@
2362     \belowdisplayskip\skip@
2363   \fi
2364 \fi
2365 }

```

TODO: describe

```

2366 \def\eq\display@leave{%
2367   \prevgraf\eq\prevgraf@
2368   \ifcase\eq\skip@mode@leave@
2369   \or
2370     \endgraf
2371   \or
2372     \endgraf
2373     \prevdepth-\@m\p@
2374 \fi
2375 }

```

TODO: describe

```

2376 \def\eq\display@nest{%
2377   \let\displaybreak\eq\displaybreak@default
2378   \let\intertext\eq\intertext@default
2379   \let\vspace\eq\vspace@org
2380 }

```

TODO: describe **TODO:** box version?! (but also consider nesting)

```

2381 \def\eq\display@restore@active{%
2382   \let\label\eq\label@org
2383   \let\tag\eq\tag@default
2384   \let\raisetag\eq\raisetag@default
2385   \let\displaybreak\eq\displaybreak@default
2386   \let\intertext\eq\intertext@default
2387   \let\vspace\eq\vspace@org
2388   \ifdefined\eq\amp@mode
2389     \let\&\eq\amp@org
2390 \fi
2391 \let\eqnpunct\eq\punct@adopt
2392 \let\eq\punct@block\@undefined
2393 \let\eq\display@restore\@empty
2394 }

```

TODO: describe

```

2395 \let\eq\display@restore\@empty
2396 \eq\append@\arrayparboxrestore{%
2397   \eq\display@restore
2398   \ifdefined\eq\ampproof@active
2399     \eq\amprevert
2400 \fi

```

```

2401 \displayfalse
2402 }

```

7.5 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)

```

2403 \def\eq\stack@enable{%
2404 \let\eq\stack@save@equations\eq\stack@save@equations@
2405 \let\eq\stack@save@box\eq\stack@save@box@
2406 }

```

TODO: describe

```

2407 \let\eq\stack@save@equations\eq\stack@enable
2408 \let\eq\stack@save@box\eq\stack@enable
2409 \let\eq\stack@restore\empty

```

TODO: describe

```

2410 \def\eq\stack@save@reg#1{\global#1\the#1\relax}
2411 \def\eq\stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

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

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

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

```

2412 \def\eq\stack@save@equations@{%
2413 \let\eq\stack@numbering@eqnswinit\eq\numbering@eqnswinit
2414 \let\eq\stack@cell@container\eq\cell@container
2415 \let\eq\stack@tags@container\eq\tags@container
2416 \let\eq\stack@interline@container\eq\interline@container
2417 \let\eq\stack@dimensions@tab\eq\dimensions@tab
2418 \let\eq\stack@block@container\eq\display@container
2419 \let\eq\stack@punct@top\eq\punct@top
2420 \edef\eq\stack@restore{%
2421 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2422 \eq\stack@save@let\eq\stack@numbering@eqnswinit\eq\numbering@eqnswinit
2423 \eq\stack@save@let\eq\stack@cell@container\eq\cell@container
2424 \eq\stack@save@let\eq\stack@tags@container\eq\tags@container
2425 \eq\stack@save@let\eq\stack@interline@container\eq\interline@container
2426 \eq\stack@save@let\eq\stack@dimensions@tab\eq\dimensions@tab
2427 \eq\stack@save@let\eq\stack@block@container\eq\display@container
2428 \eq\stack@save@let\eq\stack@punct@top\eq\punct@top
2429 \eq\stack@save@reg\eq\column@
2430 \eq\stack@save@reg\eq\totalcolumns@
2431 \eq\stack@save@reg\eq\line@avail@
2432 \eq\stack@save@reg\eq\line@pos@
2433 \eq\stack@save@reg\eq\line@width@
2434 \eq\stack@save@reg\eq\line@depth@
2435 \eq\stack@save@reg\eq\line@height@
2436 \eq\stack@save@reg\eq\line@prevdepth@
2437 \eq\stack@save@reg\eq\line@interline@
2438 \eq\stack@save@reg\eq\totalheight@
2439 \eq\stack@save@reg\eq>tagwidth@max@
2440 \eq\stack@save@reg\eq>tagpos@row@

```



```

2441 \eq\stack@save@reg\eq\row@
2442 \eq\stack@save@reg\eq\tagrows@
2443 }%
2444 }

```

TODO: describe

```

2445 \def\eq\stack@save@box@{%
2446 \let\eq\stack@cell@container\eq\cell@container
2447 \edef\eq\stack@restore{%
2448 \eq\stack@save\let\eq\stack@cell@container\eq\cell@container
2449 \eq\stack@save@reg\eq\row@
2450 }%
2451 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2452 \def\eq\measure@init#1#2{%
2453 \eq\dimensions@reset
2454 \let\eq\display@container\@empty
2455 \eq\numbering@measure@init
2456 \eq\row@\z@
2457 \eq\totalheight@\z@
2458 \eq\totalrows@\@M
2459 \eq\line@prevdepth@-\@m\p@
2460 \eq\line@interline@\z@
2461 \tabskip\z@skip
2462 \everycr{\noalign{%
2463 \global\advance\eq\row@\@ne
2464 #1%
2465 }}%
2466 \eq\punct@top@set
2467 \global\let\eq\interline@container\eq\interline@container@clear
2468 \eq\measure@savestate
2469 \eq\multi@cr@let{#2}%
2470 }

```

TODO: describe

```

2471 \def\eq\measure@tag{%
2472 \eq\tagwidth@\z@
2473 \ifdefined\eq\numbering@multi
2474 \if@eqnsw
2475 \eq\tags@container
2476 \eq\tagbox@make\eq\composetag@measure
2477 \ifdefined\eq\tagpos@reserve\else
2478 \eq\tagwidth@\z@
2479 \fi
2480 \fi
2481 \fi
2482 }

```

TODO: describe

```
2483 \def\eql@measure@endrow{%
2484   \ifdim\eql@line@prevdepth@=-\@m\p@\else
2485     \dimen@ \dimexpr\baselineskip-\eql@line@height@-\eql@line@prevdepth@\relax
2486     \ifdim\dimen@<\lineskiplimit
2487       \dimen@ \lineskip
2488     \fi
2489     \advance\eql@line@interline@\dimen@
2490   \fi
2491   \eql@dimensions@endrow
2492   \ifdim\eql@tagwidth@>\eql@tagwidth@max@
2493     \global\eql@tagwidth@max@\eql@tagwidth@
2494   \fi
2495   \ifdim\eql@tagwidth@>\z@
2496     \global\advance\eql@tagrows@\@ne
2497   \fi
2498   \global\advance\eql@totalheight@\dimexpr
2499     \eql@line@interline@+\eql@line@height@+\eql@line@depth@
2500   \global\eql@line@interline@\z@
2501   \global\eql@line@prevdepth@\eql@line@depth@
2502 }
```

TODO: describe

```
2503 \def\eql@measure@close{%
2504   \advance\eql@row@-\tw@
2505   \eql@totalrows@\eql@row@
2506   \ifnum\eql@totalrows@>\z@
2507     \eql@dimensions@get\@ne
2508     \eql@topheight@\dimexpr\eql@line@height@+\eql@line@interline@\relax
2509     \eql@dimensions@get\eql@totalrows@
2510     \eql@bottomdepth@\eql@line@depth@
2511   \fi
2512   \eql@numbering@measure@blocktag
2513   \begingroup
2514     \eql@tags@container
2515     \if@eqnsw
2516       \eql@tagbox@make\eql@composetag@measure
2517       \ifdefined\eql@tagpos@reserve\else
2518         \eql@tagwidth@\z@
2519       \fi
2520       \eql@dimensions@saveblocktag
2521     \else
2522       \eql@dimensions@savenoblocktag
2523       \eql@numbering@warnunused
2524     \fi
2525   \endgroup
2526   \eql@dimensions@get\z@
2527   \eql@measure@restorestate
2528 }
```

measure@restorestate

ql@measure@savestate

```
2529 \let\eql@measure@restorestate\@empty
2530 \def\eql@measure@savestate{%
2531   \begingroup
2532     \def\@elt##1{%
2533       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2534     \global\edef\@gtempa{\c1@@ckpt}%

```

```

2535 \endgroup
2536 \let\eql@measure@restorestate\@gtempa
2537 }

```

8.2 Line Breaks

TODO: describe

`\eql@multi@cr`

```

2538 \def\eql@multi@cr{%
2539 \let\eql@punct@term\eql@false
2540 \let\eql@class@rel@composed\@empty
2541 \eql@ampprotect\eql@multi@cr@test\eql@multi@cr@process}

```

TODO: describe

```

2542 \def\eql@multi@cr@test@setopt{%
2543 \let\eql@multi@cr@test\eql@multi@cr@testopt}
2544 \def\eql@multi@cr@test@setall{%
2545 \let\eql@multi@cr@test\eql@multi@cr@testall}

```

`\eql@multi@cr@testopt` **TODO:** describe

```

2546 \def\eql@multi@cr@testopt#1{\eql@teststaropt@tight
2547 {\eql@displaybreak@star\@M\eql@multi@cr@testopt@set{#1}}}%
2548 {\eql@multi@cr@testopt@set{#1}}{Opt}}
2549 \def\eql@multi@cr@testopt@set#1[#2]{\eql@vspace@add{#2}#1}

```

`\eql@multi@cr@testall` **TODO:** describe

`\lti@cr@testall@parse`

```

2550 \def\eql@multi@cr@testall{\eql@parseopt@cr\eql@multi@cr@testall@parse}
2551 \def\eql@multi@cr@testall@parse{%
2552 \ifx\eql@parseopt@token[%
2553 \let\eql@parseopt@next\eql@multi@cr@parse@vspace
2554 \fi
2555 \ifx\eql@parseopt@token*%
2556 \let\eql@parseopt@next\eql@multi@cr@parse@star
2557 \fi
2558 \ifx\eql@parseopt@token.%
2559 \let\eql@parseopt@next\eql@parseopt@punctpass
2560 \fi
2561 \ifx\eql@parseopt@token,%
2562 \let\eql@parseopt@next\eql@parseopt@punctpass
2563 \fi
2564 \ifx\eql@parseopt@token~%
2565 \let\eql@parseopt@next\eql@parseopt@punctpass
2566 \fi
2567 \ifx\eql@parseopt@token'%
2568 \let\eql@parseopt@next\eql@parseopt@punctnext
2569 \fi
2570 \ifx\eql@parseopt@token!%
2571 \let\eql@parseopt@next\eql@parseopt@punctterm
2572 \fi
2573 \ifx\eql@parseopt@token/%
2574 \let\eql@parseopt@next\eql@multi@cr@parse@break
2575 \fi
2576 \ifx\eql@parseopt@token=%

```

```

2577 \let\eql@parseopt@next\eql@parseopt@relsyb
2578 \fi
2579 \ifx\eql@parseopt@token;%
2580 \let\eql@parseopt@next\eql@parseopt@relcont
2581 \fi
2582 \ifx\eql@parseopt@token:%
2583 \let\eql@parseopt@next\eql@parseopt@relstart
2584 \fi
2585 \ifx\eql@parseopt@token|%
2586 \let\eql@parseopt@next\eql@parseopt@relord
2587 \fi
2588 \ifx\eql@parseopt@token?%
2589 \let\eql@parseopt@next\eql@multi@cr@parse@rel
2590 \fi
2591 \ifx\eql@parseopt@token&%
2592 \let\eql@parseopt@next\eql@parseopt@end
2593 \fi
2594 }
2595 \def\eql@multi@cr@parse@vspace[#1]{\eql@vspace@add{#1}\eql@parseopt@peek}
2596 \def\eql@multi@cr@parse@star#1{\eql@displaybreak@star\M\eql@parseopt@peek}
2597 \def\eql@multi@cr@parse@break{\numbernext\eql@parseopt@punctclear}
2598 \def\eql@multi@cr@parse@rel#1#2{%
2599 \def\eql@tmp{#2}%
2600 \ifx\eql@tmp\eql@relax\else
2601 \ifdefined\eql@punct@next\else
2602 \eql@punct@next@clear
2603 \fi
2604 \ifdefined\eql@multi@cr@relnext\numbernext\fi
2605 \fi
2606 \ifdefined\eql@multi@linesmode
2607 \ifx\eql@tmp\@empty
2608 \def\eql@class@rel@composed{\eql@shape@cont}%
2609 \else
2610 \def\eql@class@rel@composed{\eql@shape@rel#2}%
2611 \fi
2612 \else
2613 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
2614 \fi
2615 \eql@parseopt@end}

```

eql@multi@cr@process

```

2616 \def\eql@multi@cr@process{%
2617 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
2618 \edef\eql@tmp{%
2619 \unexpanded{%
2620 \eql@multi@endline
2621 \cr
2622 \eql@multi@cr@interline
2623 }%
2624 \unexpanded\expandafter{\eql@class@rel@composed}%
2625 }%
2626 \eql@tmp
2627 }

2628 \def\eql@multi@cr@interline{%
2629 \noalign{%
2630 \eql@interline@container
2631 \eql@display@injectbefore

```

```

2632 \ifnum\eqldisplaybreak@pen@=\@MM
2633 \penalty\interdisplaylinepenalty
2634 \else
2635 \penalty\eqldisplaybreak@pen@
2636 \fi
2637 \vskip\eqlvspaceskip@
2638 \global\advance\eqlline@interline@\eqlvspaceskip@
2639 \eqldisplay@injectafter
2640 \global\let\eql@interline@container\eql@interline@container@clear
2641 }%
2642 }

```

`\eql@multi@cr@let`

```

2643 \def\eql@multi@cr@let#1{%
2644 \let\\eql@multi@cr
2645 \let\eql@multi@endline#1%
2646 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2647 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2648 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2649 \def\eql@intertext@process{%
2650 \eql@multi@endline
2651 \cr
2652 \ifmeasuring@
2653 \expandafter\@gobble
2654 \else
2655 \expandafter\eql@intertext@print
2656 \fi
2657 }

```

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

```

2658 \def\eql@intertext@print#1{%
2659 \noalign{%
2660 \eqldisplay@halign@end
2661 \let\eql@skip@force@below\z@
2662 \let\eql@skip@force@above\z@
2663 \eql@setkeys{intertext}\eql@intertext@opt
2664 \openup-\eql@spread@
2665 \penalty\postdisplaypenalty
2666 \ifcase\eql@skip@force@below\relax
2667 \advance\eqlvspaceskip@\glueexpr\eql@skip@long@below\relax
2668 \or
2669 \advance\eqlvspaceskip@\glueexpr\eql@skip@short@below\relax
2670 \or
2671 \advance\eqlvspaceskip@\glueexpr\eql@skip@cont@below\relax
2672 \or
2673 \advance\eqlvspaceskip@\glueexpr\eql@skip@par@below\relax
2674 \or

```

```

2675     \advance\eq\vspace\glue\expr\eq\skip\top\below\relax
2676 \or
2677     \advance\eq\vspace\z\skip
2678 \or
2679     \advance\eq\vspace\glue\expr\eq\skip\med\below\relax
2680 \or
2681     \advance\eq\vspace\glue\expr\eq\skip\custom\below\relax
2682 \fi
2683 \vskip\eq\vspace
2684 \global\let\eq\interline\container\eq\interline\container\clear
2685 \vbox{%
2686     \parboxrestore
2687     \ifdim
2688         \ifdim\totalleftmargin=\z\linewidth\else-\maxdimen\fi=\columnwidth
2689     \else
2690         \parshape\@ne
2691         \totalleftmargin\linewidth
2692     \fi
2693     \noindent
2694     \prevgraf\eq\prevgraf
2695     \ignorespaces
2696     #1%
2697     \par
2698     \global\eq\prevgraf\prevgraf
2699 }%
2700 \penalty\predisplaypenalty
2701 \ifcase\eq\skip\force\above\relax
2702     \vskip\glue\expr\eq\skip\long\above\relax
2703 \or
2704     \vskip\glue\expr\eq\skip\short\above\relax
2705 \or
2706     \vskip\glue\expr\eq\skip\cont\above\relax
2707 \or
2708     \vskip\glue\expr\eq\skip\par\above\relax
2709 \or
2710     \vskip\glue\expr\eq\skip\top\above\relax
2711 \or
2712     \vskip\z\skip
2713 \or
2714     \vskip\glue\expr\eq\skip\med\above\relax
2715 \or
2716     \vskip\glue\expr\eq\skip\custom\above\relax
2717 \fi
2718 % \eq\prevdepth\maxdimen
2719 \eq\prevdepth\z
2720 \eq\display\halign\start
2721 }
2722 }

```

TODO: describe

```

2723 \newenvironment{eq\intertext}{%
2724     \eq\testopt\tight\eq\intertext{%
2725 }{%
2726     \aftergroup\eq\intertext\after
2727     \ignorespacesafterend
2728 }

```

TODO: describe

```

2729 \def\eql@intertext@env{intertext}
2730 \def\eql@intertext@[#1]{%
2731   \global\def\eql@intertext@opt{#1}%
2732   \ifx\@currenvir\eql@intertext@env
2733     \def\eql@scan@call{\eql@intertext@inject\eql@scan@end}%
2734     \expandafter\eql@scan@env
2735   \else
2736     \expandafter\eql@intertext@process
2737   \fi
2738 }

```

TODO: describe

```

2739 \def\eql@intertext@inject{%
2740   \global\edef\eql@intertext@after{%
2741     \noexpand\eql@intertext@process{%
2742       \ifx\eql@scan@body\eql@scan@body@dump
2743         \eql@scan@body@dump
2744       \else
2745         \noexpand\scantokens{\eql@scan@body@dump}%
2746       \fi
2747     }%
2748   }%
2749 }

```

8.4 Line Marks

TODO: describe

```

2750 \def\eql@markline@pos@below{below}
2751 \def\eql@markline@pos@bottom{bottom}
2752 \def\eql@markline@pos@baseline{baseline}
2753 \let\eql@markline@pos\eql@markline@pos@baseline
2754 \let\eql@markline@shift\z@
2755 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2756 \def\eql@markline@symbol{ }

```

TODO: describe

```

2757 \def\eql@markline@select#1{%
2758   \let\eql@markline@shift\z@
2759   \eql@setkeys{markline}{#1}%
2760   \eql@markline@print
2761 }

```

TODO: describe

```

2762 \def\eql@markline@print{%
2763   \dimen@ \dimexpr\eql@markline@shift\relax
2764   \ifx\eql@markline@pos\eql@markline@pos@below
2765     \ifdim\dimen@=\z@\else
2766       \penalty\@M
2767       \vskip-\dimen@
2768     \fi
2769     \nointerlineskip
2770     \penalty\@M
2771     \vbox{\hfill\hbox{\eql@markline@symbol}}%
2772   \else
2773     \ifx\eql@markline@pos\eql@markline@pos@baseline

```

```

2774     \advance\dimen@\prevdepth
2775     \fi
2776     \setbox\z@\hbox{\raise\dimen@\hbox{\eq@markline@symbol}}}%
2777     \dimen@\prevdepth
2778     \ht\z@\z@
2779     \dp\z@\z@
2780     \nointerlineskip
2781     \penalty\@M
2782     \vbox{\hfill\box\z@}%
2783     \prevdepth\dimen@
2784 \fi
2785 }

```

TODO: describe

```

2786 \def\eq@markline@inject#1{%
2787   \let\eq@markline@push\eq@false
2788   \ifx\eq@markline@pos\eq@markline@pos@below\else
2789     \ifdefined\eq@tagsleft\else
2790       \ifx\eq@equations@main\eq@multi@main
2791         \ifdefined\eq@numbering@multi
2792           \if@eqnsw
2793             \let\eq@markline@push\eq@true
2794           \fi
2795         \else
2796           \ifnum\eq@row@=\eq@tagpos@row@
2797             \let\eq@markline@push\eq@true
2798           \fi
2799         \fi
2800       \else
2801         \if@eqnsw
2802           \let\eq@markline@push\eq@true
2803         \fi
2804       \fi
2805     \fi
2806   \fi
2807   \ifdefined\eq@markline@push
2808     \global\eq@append\eq@interline@container{%
2809       \eq@append\eq@display@injectbefore{\eq@markline@select{push,#1}}}%
2810   \else
2811     \global\eq@append\eq@interline@container{%
2812       \eq@append\eq@display@injectbefore{\eq@markline@select{#1}}}%
2813   \fi
2814 }

```

TODO: describe

```

2815 \def\eq@markline@amsthm@opt[#1]{\eq@markline@inject{qed,#1}}
2816 \def\eq@markline@amsthm@staropt[#1]{\eq@markline@inject{qed,push,#1}}
2817 \def\eq@markline@amsthm@qed{\eq@teststaropt@tight
2818   \eq@markline@amsthm@staropt\eq@markline@amsthm@opt{}}
2819 \def\eq@markline@amsthm@register#1{\eq@letcs{#1@qed}\eq@markline@amsthm@qed}
2820 \def\eq@markline@amsthm@move#1#2{%
2821   \AddToHook{package/amsthm/after}{%
2822     \eq@letcs{#1@qed}\expandafter\cename#2@qed\endcsname}}

```


9 Column Placement

TODO: describe

9.1 Supporting Definitions

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

```
2823 \newcount\eq@shape@pos@
2824 \newdimen\eq@shape@amount@
2825 \let\eq@shape@lastrow\eq@false
```

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

`\eq@marginright@` (*dimen*)
`\eq@marginleft@min@` (*dimen*)
`\eq@centeroffset@` (*dimen*)

```
2826 \newdimen\eq@marginleft@
2827 \newdimen\eq@marginright@
2828 \newdimen\eq@marginleft@min@
2829 \newdimen\eq@centeroffset@
```

9.2 Shape Schemes

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

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

```
2830 \def\eq@shape@tab@default{default}
2831 \def\eq@shape@tab@left{left}
2832 \def\eq@shape@tab@center{center}
2833 \def\eq@shape@tab@right{right}
2834 \def\eq@shape@tab@first{first}
2835 \def\eq@shape@tab@hanging{hanging}
2836 \def\eq@shape@tab@steps{steps}
```

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

```
2837 \let\eq@shape@tab@def\eq@shape@tab@default
2838 \let\eq@shape@tab@\eq@shape@tab@default
2839 \let\eq@shape@tab@l\eq@shape@tab@left
2840 \let\eq@shape@tab@c\eq@shape@tab@center
2841 \let\eq@shape@tab@r\eq@shape@tab@right
2842 \let\eq@shape@tab@rc\eq@shape@tab@first
2843 \let\eq@shape@tab@indent\eq@shape@tab@first
2844 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2845 \let\eq@shape@tab@lc\eq@shape@tab@hanging
2846 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2847 \let\eq@shape@tab@lcr\eq@shape@tab@steps
```

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

```
2848 \let\eq@shape@mode\eq@shape@tab@default
```

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

```
2849 \def\eq@shape@set#1{%
```

```

2850 \ifcsname eql@shape@tab@#1\endcsname
2851   \expandafter\let\expandafter\eql@shape@mode
2852   \csname eql@shape@tab@#1\endcsname
2853 \else
2854   \eqLError{shape '#1' unknown: setting to default}%
2855   \let\eql@shape@mode\eql@shape@tab@default
2856 \fi
2857 }

```

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

```

2858 \def\eql@shape@layoutcenter@left{\eql@shape@pos@z@eql@shape@amount@z@}
2859 \def\eql@shape@layoutcenter@center{\eql@shape@pos@ne\eql@shape@amount@z@}
2860 \def\eql@shape@layoutcenter@right{\eql@shape@pos@tw@eql@shape@amount@z@}
2861 \let\eql@shape@layoutcenter@default\eql@shape@layoutcenter@center
2862 \def\eql@shape@layoutleft@left{\eql@shape@pos@z@eql@shape@amount@z@}
2863 \def\eql@shape@layoutleft@center{\eql@shape@pos@ne\eql@shape@amount@z@}
2864 \def\eql@shape@layoutleft@right{\eql@shape@pos@tw@eql@shape@amount@z@}
2865 \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):

```

2866 \def\eql@shape@layoutcenter@first{%
2867   \eql@shape@pos@z@
2868   \eql@shape@amount@z@
2869   \ifnum\eql@totalrows@>\@ne
2870     \ifnum\eql@row@=\@ne
2871       \eql@shape@amount@\eql@indent@
2872     \fi
2873   \fi
2874 }
2875 \def\eql@shape@layoutleft@first{%
2876   \eql@shape@pos@z@
2877   \eql@shape@amount@z@
2878   \ifnum\eql@totalrows@>\@ne
2879     \ifnum\eql@row@=\@ne
2880       \eql@shape@amount@\eql@indent@
2881     \fi
2882   \fi
2883 }

```

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:

```

2884 \def\eql@shape@layoutcenter@hanging{%
2885   \eql@shape@pos@z@
2886   \eql@shape@amount@\eql@indent@
2887   \ifnum\eql@totalrows@>\@ne
2888     \ifnum\eql@row@=\@ne
2889       \eql@shape@amount@z@
2890     \fi
2891   \fi
2892 }
2893 \def\eql@shape@layoutleft@hanging{%
2894   \eql@shape@pos@z@
2895   \eql@shape@amount@z@

```

```

2896 \ifnum\eq@totalrows@>\@ne
2897   \ifnum\eq@row@=\@ne
2898     \eq@shape@amount@-\eq@indent@
2899   \fi
2900 \fi
2901 }

```

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:

```

2902 \def\eq@shape@layoutcenter@steps{%
2903   \eq@shape@amount@z@
2904   \eq@shape@pos@\@ne
2905   \ifnum\eq@totalrows@>\@ne
2906     \ifnum\eq@row@=\@ne
2907       \eq@shape@pos@z@
2908     \fi
2909     \ifnum\eq@row@=\eq@totalrows@
2910       \eq@shape@pos@tw@
2911     \fi
2912   \fi
2913 }
2914 \def\eq@shape@layoutleft@steps{%
2915   \eq@shape@pos@z@
2916   \eq@shape@amount@z@
2917   \ifnum\eq@totalrows@>\@ne
2918     \ifnum\eq@row@=\@ne
2919       \eq@shape@amount@-\eq@indent@
2920     \fi
2921     \ifnum\eq@row@=\eq@totalrows@
2922       \eq@shape@amount@\eq@indent@
2923     \fi
2924   \fi
2925 }

```

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

```

2926 \let\eq@shape@eval\undefined
2927 \def\eq@shape@select{%
2928   \expandafter\let\expandafter\eq@shape@eval
2929     \csname eq@shape%
2930       @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2931       @\eq@shape@mode\endcsname
2932 }

```

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

`\eq@shape@alignright`

`\eq@shape@aligncenter`

```

2933 \protected\def\eq@shape@alignleft{%
2934   \global\eq@append\eq@cell@container{\eq@shape@pos@z@}%
2935   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2936 \protected\def\eq@shape@aligncenter{%
2937   \global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}%
2938   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2939 \protected\def\eq@shape@alignright{%
2940   \global\eq@append\eq@cell@container{\eq@shape@pos@tw@}%
2941   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}

```

```

2942 \def\eq@shape@align@testpar#1{%
2943   \eq@ifstar@tight{#1[\eq@indent@]}%
2944   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2945   {\eq@testopt@tight{#1}\z@}}
2946 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2947 \protected\def\eq@shape@alignamount{%
2948   \eq@ampprotecttwo\eq@ifstar@tight
2949   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2950 \def\eq@shape@alignamount@add#1{%
2951   \global\eq@appendexpand\eq@cell@container{%
2952     \advance\eq@shape@amount@the\glueexpr#1\relax\relax}}
2953 \def\eq@shape@alignamount@set#1{%
2954   \global\eq@appendexpand\eq@cell@container{%
2955     \eq@shape@amount@the\glueexpr#1\relax\relax}}
2956 \def\eq@shape@align@enable{%
2957   \let\shoveleft\eq@shape@alignleft
2958   \let\shovecenter\eq@shape@aligncenter
2959   \let\shoveright\eq@shape@alignright
2960   \let\shoveby\eq@shape@alignamount
2961 }

```

TODO: describe

```

2962 \protected\def\eq@shape@align@default{%
2963   \eq@warn@here{\shove...}%
2964   \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2965 \protected\def\eq@shape@alignamount@default{%
2966   \eq@warn@here{\shove...}%
2967   \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2968 \def\eq@shape@align@disable{%
2969   \let\shoveleft\eq@shape@align@default
2970   \let\shovecenter\eq@shape@align@default
2971   \let\shoveright\eq@shape@align@default
2972   \let\shoveby\eq@shape@alignamount@default
2973 }

```

9.3 Width Data

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

```

2974 \newdimen\eq@tagwidth@block@
2975 \newdimen\eq@tagheight@block@
2976 \newdimen\eq@tagdepth@block@

```

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

```

2977 \let\eq@dimensions@tab\@empty

```

`\eq@dimensions@reset`

```

2978 \def\eq@dimensions@reset{%
2979   \let\eq@dimensions@tab\@empty
2980   \eq@tagwidth@max@\z@
2981   \eq@tagrows@\z@
2982 }

```

\eqldimensions@add

```
2983 \def\eqldimensions@add#1{%
2984   \global\eql@appendexpand\eqldimensions@tab{#1}%
2985 }
```

eqldimensions@addreg

```
2986 \def\eqldimensions@addreg#1{#1\the#1\relax}
```

@dimensions@startrow

```
2987 \def\eqldimensions@startrow{%
2988   \eqldimensions@add{\eqldimensions@addreg\eqldimensions@row}%
2989 }
```

@dimensions@savecell

```
2990 \def\eqldimensions@savecell{%
2991   \eqldimensions@add{%
2992     \eqldimensions@addreg\eqldimensions@shape@pos@
2993     \eqldimensions@addreg\eqldimensions@cellwidth@
2994     \eqldimensions@addreg\eqldimensions@shape@amount@
2995     \noexpand\eqldimensions@cellcall
2996   }%
2997 }
```

ldimensions@savesep

```
2998 \def\eqldimensions@savesep{%
2999   \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
3000 }
```

eqldimensions@endrow

```
3001 \def\eqldimensions@endrow{%
3002   \eqldimensions@add{%
3003     \eqldimensions@addreg\eqldimensions@tagwidth@
3004     \eqldimensions@addreg\eqldimensions@line@height@
3005     \eqldimensions@addreg\eqldimensions@line@depth@
3006     \eqldimensions@addreg\eqldimensions@line@interline@
3007   };}%
3008 }
```

ensions@saveblocktag

```
3009 \def\eqldimensions@saveblocktag{%
3010   \eqldimensions@add{\eqldimensions@row@0\relax,%
3011     \eqldimensions@tagwidth@block@\the\eqldimensions@tagwidth@\relax
3012     \eqldimensions@tagheight@block@\the\eqldimensions@tagheight@\relax
3013     \eqldimensions@tagdepth@block@\the\eqldimensions@tagdepth@\relax
3014     \eqldimensions@tagpos@block@\the\eqldimensions@tagpos@\relax
3015     \let\noexpand\eqldimensions@tagpos@reserve\ifdefined\eqldimensions@tagpos@reserve
3016     \noexpand\eqldimensions@tagpos@true\else\noexpand\eqldimensions@tagpos@false\fi
3017   };}%
3018   \global\eqldimensions@tagwidth@max@\eqldimensions@tagwidth@
3019   \global\eqldimensions@tagrows@\one
3020 }
```

sions@savenoblocktag

```
3021 \def\eqldimensions@savenoblocktag{%
3022   \eqldimensions@add{\eqldimensions@row@0\relax,;}%
3023 }
```

\eqldimensions@for

```
3024 \def\eqldimensions@for#1{%
3025   \def\eqldimensions@forcall{#1}%
3026   \expandafter\eqldimensions@forstep\eqldimensions@tab
3027 }
```

l@dimensions@forstep

```
3028 \def\eqldimensions@forstep\eqldimensions@row@#1\relax#2,#3{%
3029   \eqldimensions@row@#1\relax
3030   \ifnum\eqldimensions@row@#1=\z@ \else
3031     #3%
3032     \def\eqldimensions@cells{#2}%
3033     \eqldimensions@forcall
3034     \expandafter\eqldimensions@forstep
3035   \fi
3036 }
```

\eqldimensions@get

```
3037 \def\eqldimensions@get#1{%
3038   \eqldimensions@row@#1\relax
3039   \expandafter\eqldimensions@getdef\expandafter{\the\eqldimensions@}%
3040   \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil
3041 }
```

ql@dimensions@getdef

```
3042 \def\eqldimensions@getdef#1{%
3043   \def\eqldimensions@getparse
3044     ##1\eqldimensions@row@#1\relax##2,##3;##4\@nil{%
3045     ##3%
3046     \def\eqldimensions@cells{##2}%
3047   }%
3048 }
```

\eqldimensions@colwidth@tab

```
3049 \let\eqldimensions@colwidth@tab\empty
```

\eqldimensions@colwidth@get

```
3050 \def\eqldimensions@colwidth@get#1{%
3051   \ifcase\expandafter#1\eqldimensions@colwidth@tab\else\z@ \fi
3052 }
```

\eqldimensions@colwidth@save

```
3053 \def\eqldimensions@colwidth@save#1{%
3054   \edef\eqldimensions@colwidth@tab{%
3055     \noexpand\or\the#1%
3056     \unexpanded\expandafter{\eqldimensions@colwidth@tab}%
3057   }%
3058 }
```

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

```

3059 \def\eqldimensions@calc{%
3060   \eqldcolumn@\z@
3061   \eqldline@pos@\z@
3062   \eqldline@possep@\z@
3063   \eqldline@avail@\eqldtotalwidth@
3064   \eqldline@availsep@\eqldintercolumns@
3065   \eqldline@width@\z@
3066   \eqldline@widthsep@\z@
3067   \let\eqldimensions@cellcall\eqldimensions@calc@call
3068   \let\eqldimensions@sepcall\eqldimensions@calc@callsep
3069   \eqldimensions@cells
3070 }

```

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

```

3071 \def\eqldimensions@calc@callsep{%
3072   \advance\eqldline@possep@\@ne
3073 }%

```

`\eqldimensions@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?!)

```

3074 \def\eqldimensions@calc@call{%
3075   \advance\eqldcolumn@\@ne
3076   \ifnum\eqldtotalcolumns@=\@ne
3077     \dimen@\eqldtotalwidth@
3078   \else
3079     \dimen@\eqldcolwidth@get\eqldcolumn@\relax
3080   \fi
3081   \ifdim\eqldcellwidth@>\z@
3082     \ifdim\eqldline@width@=\z@
3083       \eqldline@avail@\eqldline@pos@
3084       \eqldline@availsep@\eqldline@possep@
3085       \ifcase\eqldshape@pos@
3086       \or
3087         \advance\eqldline@avail@\dimexpr
3088           (\dimen@-\eqldcellwidth@+\eqldcenteroffset@)/\tw@\relax
3089       \or
3090         \advance\eqldline@avail@\dimexpr\dimen@-\eqldcellwidth@\relax
3091       \fi
3092       \advance\eqldline@avail@\eqldshape@amount@
3093     \fi
3094     \eqldline@width@\eqldline@pos@
3095     \eqldline@widthsep@\eqldline@possep@
3096     \ifcase\eqldshape@pos@
3097     \advance\eqldline@width@\eqldcellwidth@
3098     \or
3099     \advance\eqldline@width@\dimexpr
3100       (\dimen@+\eqldcellwidth@+\eqldcenteroffset@)/\tw@\relax
3101     \or
3102     \advance\eqldline@width@\dimen@

```

```

3103     \fi
3104     \advance\eql@line@width@\eql@shape@amount@
3105 \fi
3106 \advance\eql@line@pos@\dimen@
3107 }

```

9.4 Best Line Selection

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

```

3108 \let\eql@numbering@best@auto\eql@false

```

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

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

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

```

3109 \newcount\eql@numbering@best@row@
3110 \newdimen\eql@numbering@best@space@
3111 \let\eql@numbering@best@use\eql@false

```

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

```

3112 \def\eql@numbering@best@find{%
3113   \eql@numbering@best@row@ \z@
3114   \eql@numbering@best@space@ \z@
3115   \eql@dimensions@for{%
3116     \eql@dimensions@calc
3117     \ifdefined\eql@tagsleft
3118       \dimen@\eql@line@avail@
3119     \else
3120       \dimen@\dimexpr\eql@totalwidth@-\eql@line@width@\relax
3121     \fi
3122     \ifdim\dimen@>\eql@numbering@best@space@
3123       \eql@numbering@best@row@\eql@row@
3124       \eql@numbering@best@space@\dimen@
3125     \fi
3126   }%
3127   \ifnum\eql@numbering@best@row@>\z@
3128     \eql@tagpos@row@\eql@numbering@best@row@
3129     \let\eql@tagpos@continuous\eql@false
3130     \eql@tagpos@prevrow@\z@
3131   \fi
3132 }

```

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

```

3133 \def\eql@numbering@best@test#1{%
3134   \eql@dimensions@get#1%
3135   \eql@dimensions@calc
3136   \ifdefined\eql@tagsleft
3137     \dimen@\dimexpr\eql@line@avail@
3138       +\eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax
3139   \else
3140     \dimen@\dimexpr\displaywidth-\eql@line@width@
3141       -\eql@marginleft@-\eql@line@widthsep@\eql@colsep@\relax
3142   \fi
3143   \ifdim\dimen@<\eql@tagwidth@block@
3144     \let\eql@numbering@best@use\eql@true
3145   \fi
3146 }

```


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

```

3147 \def\eq@numbering@best@eval{%
3148   \ifdefined\eq@numbering@best@auto
3149     \ifdefined\eq@numbering@best@use\else
3150       \ifdefined\eq@numbering@multi\else
3151         \ifnum\eq@tagpos@row@>\z@
3152           \eq@numbering@best@test\eq@tagpos@row@
3153         \fi
3154         \ifnum\eq@tagpos@prevrow@>\z@
3155           \eq@numbering@best@test\eq@tagpos@prevrow@
3156         \fi
3157       \fi
3158     \fi
3159   \fi
3160   \ifdefined\eq@numbering@best@use
3161     \eq@numbering@best@find
3162   \fi
3163 }

```

9.5 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`

```

3164 \def\eq@adjust@calc@tagmargin{%
3165   \ifdefined\eq@tagmargin@val
3166     \eq@tagmargin@\glueexpr\eq@tagmargin@val\relax
3167   \else
3168     \eq@tagmargin@\eq@tagwidth@max@
3169     \ifdim\eq@tagmargin@>\z@
3170       \advance\eq@tagmargin@-\eq@tagsepmin@
3171     \fi
3172   \fi

3173   \dimen@\eq@tagrows@\p@
3174   \ifnum\eq@totalrows@=\@ne
3175     \ifnum\eq@tagrows@=\@ne
3176       \advance\dimen@1sp\relax
3177     \fi
3178   \fi
3179   \ifdim\dimen@>\eq@totalrows@\eq@tagmargin@ratio@\else
3180     \eq@tagmargin@\z@
3181   \fi

3182   \@tempdima\dimexpr\displaywidth
3183     -\eq@totalwidth@-\eq@intercolumns@\eq@colsepmin@\relax
3184   \@tempdimb\dimexpr\@tempdima-\tw@\eq@tagmargin@\relax
3185   \ifdim\@tempdimb>\z@
3186     \ifdim\eq@tagmargin@threshold\@tempdima<\@tempdimb
3187       \eq@tagmargin@\z@
3188     \fi
3189   \fi
3190 }

```

9.6 Single Column

ql@adjust@calc@lines

```

3191 \def\eq@adjust@calc@lines{%
3192   \eq@totalcolumns@\ne
3193   \eq@intercolumns@\z@
3194   \eq@colsep@\z@
3195   \ifdefined\eq@layoutleft
3196     \eq@marginleft@\glueexpr\eq@layoutleftmargin\relax
3197     \eq@marginleft@min@\glueexpr\eq@layoutleftmarginmin\relax
3198     \ifdim\eq@marginleft@<\eq@marginleft@min@
3199       \eq@marginleft@\eq@marginleft@min@
3200     \fi
3201     \dimen@\glueexpr\eq@layoutleftmarginmax\relax
3202     \ifdim\eq@marginleft@>\dimen@
3203       \eq@marginleft@\dimen@
3204     \fi
3205     \eq@marginright@\z@
3206     \eq@centeroffset@\z@
3207   \else
3208     \eq@adjust@calc@tagmargin
3209     \ifdefined\eq@paddingleft@val
3210       \eq@marginleft@\dimexpr
3211         (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3212         -\glueexpr\eq@paddingleft@val\relax\relax
3213       \ifdim\eq@marginleft@<\z@
3214         \eq@marginleft@\z@
3215       \fi
3216     \else
3217       \eq@marginleft@\z@
3218     \fi
3219     \ifdefined\eq@paddingright@val
3220       \eq@marginright@\dimexpr
3221         (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3222         -\glueexpr\eq@paddingright@val\relax\relax
3223       \ifdim\eq@marginright@<\z@
3224         \eq@marginright@\z@
3225       \fi
3226     \else
3227       \eq@marginright@\z@
3228     \fi
3229     \ifdim\eq@tagmargin@>\z@
3230       \ifdefined\eq@tagsleft
3231         \ifdim\eq@marginleft@<\eq@tagsepmin@
3232           \eq@marginleft@\eq@tagsepmin@
3233         \fi
3234         \advance\eq@marginleft@\eq@tagmargin@
3235         \advance\eq@centeroffset@\eq@tagmargin@
3236       \else
3237         \ifdim\eq@marginright@<\eq@tagsepmin@
3238           \eq@marginright@\eq@tagsepmin@
3239         \fi
3240         \advance\eq@marginright@\eq@tagmargin@
3241         \advance\eq@centeroffset@-\eq@tagmargin@
3242       \fi
3243     \fi
3244     \eq@marginleft@min@\z@
3245     \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@

```

```

3246         \ifdefined\eql@tagsleft+\else-\fi\eql@tagmargin@relax
3247     \fi

3248     \eql@totalwidth@\dimexpr\displaywidth
3249         -\eql@marginleft@-\eql@marginright@relax
3250 }

```

9.7 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:

```

3251 \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:

```

3252     \ifnum\eql@totalcolumns@<\thr@@
3253         \eql@totalcolumns@\tw@
3254         \let\eql@columns@fulllength\eql@false
3255     \fi

```

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

```

3256     \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@relax

```

Evaluate the minimum intercolumn space which we will need often:

```

3257     \eql@colsepmin@\glueexpr\eql@colsepmin@valrelax

```

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

```

3258     \ifdefined\eql@layoutleft
3259         \eql@marginleft@\glueexpr\eql@layoutleftmargin@relax
3260         \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin@relax
3261         \ifdim\eql@marginleft@<\eql@marginleft@min@
3262             \eql@marginleft@\eql@marginleft@min@
3263         \fi
3264     \else

```

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

```

3265         \eql@adjust@calc@tagmargin
3266         \ifdefined\eql@columns@fulllength
3267             \ifdim\eql@tagmargin@>\z@
3268                 \advance\eql@tagmargin@\eql@tagsepmin@
3269             \fi
3270         \fi
3271         \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
3272             -\eql@intercolumns@\eql@colsepmin@relax
3273             \eql@tagmargin@\z@
3274         \fi
3275         \eql@marginleft@min@\z@
3276     \fi

```

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

```
3277 \ifnum\eq@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:

```
3278 \eq@colsep@\dimexpr\displaywidth-\eq@totalwidth@\relax
3279 \ifdefined\eq@layoutleft
3280 \advance\eq@colsep@-\eq@marginleft@
3281 \else
3282 \advance\eq@colsep@-\eq@tagmargin@
3283 \fi
3284 \count@\eq@intercolumns@
3285 \ifdefined\eq@columns@fulllength\else
3286 \ifdefined\eq@layoutleft
3287 \advance\count@\@ne
3288 \else
3289 \advance\count@\tw@
3290 \fi
3291 \fi
3292 \divide\eq@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:

```
3293 \ifdim\eq@colsep@<\eq@colsepmin@
3294 \eq@colsep@\eq@colsepmin@
3295 \else
3296 \ifdefined\eq@columns@fulllength\else
3297 \dimen@\glueexpr\eq@colsepmax@val\relax
3298 \ifdim\eq@colsep@>\dimen@
3299 \eq@colsep@\dimen@
3300 \fi
3301 \fi
3302 \fi
3303 \else
```

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

```
3304 \eq@colsep@\eq@colsepmin@
3305 \fi
```

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

```
3306 \ifdefined\eq@layoutleft
```

Set the default value:

```
3307 \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:

```
3308 \dimen@\dimexpr\displaywidth-\eq@totalwidth@
3309 -\eq@intercolumns@\eq@colsep@\relax
3310 \ifdim\dimen@<\eq@marginleft@
3311 \ifdim\dimen@<\eq@marginleft@min@
3312 \eq@marginleft@\eq@marginleft@min@
3313 \else
```

```

3314         \eql@marginleft@\dimen@
3315     \fi
3316 \fi
3317 \fi
3318 \else

```

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

```

3319     \ifdefined\eql@columns@fulllength
3320         \eql@marginleft@z@

```

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

```

3321 \else
3322     \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@
3323         -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
3324     \ifdim\eql@marginleft@<z@
3325         \eql@marginleft@z@
3326     \fi
3327 \fi

```

Add tag margin in case of left tags:

```

3328     \ifdefined\eql@tagsleft
3329         \advance\eql@marginleft@\eql@tagmargin@
3330     \fi
3331 \fi

```

Find the best row for tag placement:

```

3332 \eql@numbering@best@eval

```

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

```

3333 \ifdefined\eql@tagsleft
3334     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
3335 \else
3336     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
3337 \fi

```

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

```

3338 \ifdefined\eql@numbering@multi
3339     \eql@dimensions@for{%
3340         \ifdim\eql@tagwidth@>z@
3341             \eql@dimensions@calc
3342             \eql@adjust@columns@test
3343         \fi
3344     }%
3345 \else
3346     \ifnum\eql@tagpos@row@>z@
3347         \ifnum\eql@tagpos@row@>\eql@totalrows@\else
3348             \eql@dimensions@get\eql@tagpos@row@
3349             \eql@tagwidth@\eql@tagwidth@block@
3350             \eql@dimensions@calc
3351             \eql@adjust@columns@test
3352         \fi

```

```

3353 \fi
3354 \ifnum\eq@tagpos@prevrow@>\z@
3355 \eq@dimensions@get\eq@tagpos@prevrow@
3356 \eq@tagwidth@\eq@tagwidth@block@
3357 \eq@dimensions@calc
3358 \eq@adjust@columns@test
3359 \fi
3360 \fi

```

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

```

3361 \advance\eq@totalwidth@\dimexpr
3362 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
3363 }

```

Placement for Right Tags.

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

```

3364 \def\eq@adjust@columns@test@tagsright{%

```

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

```

3365 \@tempdima\dimexpr\displaywidth-\eq@line@width@-\eq@tagwidth@\relax

```

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

```

3366 \ifdim\@tempdima<\dimexpr
3367 \eq@marginleft@+\eq@line@widthsep@\eq@colsep@\relax

```

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

```

3368 \ifdim\@tempdima<\dimexpr
3369 \eq@marginleft@min@+\eq@line@widthsep@\eq@colsepmin@\relax\else

```

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

```

3370 \eq@adjust@columns@modify@tagsright
3371 \fi
3372 \fi
3373 }

```

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

```

3374 \def\eq@adjust@columns@modify@tagsright{%

```

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

```

3375 \ifnum\eq@line@widthsep@>\z@
3376 \dimen@\@tempdima
3377 \count@\eq@line@widthsep@
3378 \ifdefined\eq@layoutleft
3379 \advance\dimen@-\eq@marginleft@

```

```

3380 \else
3381 \ifdefined\eql@columns@fulllength\else
3382 \advance\count@\@ne
3383 \fi
3384 \fi
3385 \divide\dimen@\count@

```

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

```

3386 \ifdim\dimen@<\eql@colsep@
3387 \ifdim\dimen@<\eql@colsepmin@
3388 \eql@colsep@\eql@colsepmin@
3389 \else
3390 \eql@colsep@\dimen@
3391 \fi
3392 \fi
3393 \fi

```

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

```

3394 \dimen@\dimexpr\@tempdima-\eql@line@widthsep@\eql@colsep@\relax
3395 \ifdim\eql@marginleft@>\dimen@
3396 \eql@marginleft@\dimen@
3397 \fi
3398 }

```

Placement for Left Tags.

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

```

3399 \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.

```

3400 \count@\numexpr\eql@intercolumns@-\eql@line@availsep@\relax
3401 \@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.

```

3402 \ifdim\@tempdima>\dimexpr
3403 \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).

```

3404 \ifdim\eql@tagwidth@<%
3405 \ifdefined\eql@layoutleft
3406 \glueexpr\eql@layoutleftmarginmax\relax
3407 \else
3408 \displaywidth
3409 \fi

```

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

```

3410 \ifdim\@tempdima>\dimexpr
3411 \displaywidth-\eql@totalwidth@-\count@\eql@colsepmin@\relax\else

```

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

```

3412         \eql@adjust@columns@modify@tagsleft
3413     \fi
3414 \fi
3415 \fi
3416 }

```

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

```

3417 \def\eql@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.

```

3418 \ifnum\count@>\z@
3419     \dimen@ \dimexpr\displaywidth-\eql@totalwidth@-\@tempdima\relax
3420     \ifdefined\eql@columns@fulllength\else
3421         \advance\count@\@ne
3422     \fi
3423     \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.

```

3424     \ifdim\dimen@<\eql@colsep@
3425         \ifdim\dimen@<\eql@colsepmin@
3426             \dimen@\eql@colsepmin@
3427         \fi
3428         \advance\dimen@-\eql@colsep@
3429         \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3430         \advance\eql@colsep@\dimen@
3431     \fi
3432 \fi

```

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

```

3433 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3434 \ifdim\eql@marginleft@<\dimen@
3435     \eql@marginleft@\dimen@
3436 \fi
3437 }

```

10 Single Column Arrangement

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

10.1 Supporting Definitions

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

```

3438 \ifdefined\inf@bad\else%
3439     \newcount\inf@bad
3440     \inf@bad1000000\relax
3441 \fi

```


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

```
3442 \let\eq restore@hfuzz\empty
3443 \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 `\eq tagbadness@` threshold for shrinking equation lines to the intended margin or to fit into the line at all before the tag is raised or lowered:

```
3444 \newcount\eq alignbadness@
3445 \newcount\eq tagbadness@
3446 \newcount\eq arrange@badness@
3447 \eq alignbadness@\inf@bad
3448 \eq tagbadness@\inf@bad
```

10.2 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@`:

```
3449 \def\eq arrange@try#1{%
3450   \ifdim#1>\dimexpr\displaywidth-\eq cellwidth@\relax
3451     \setbox\eq cellbox@\hbox to\displaywidth{%
3452       \unhbox\eq cellbox@\unkern\kern#1}%
3453     \eq arrange@badness@\badness
3454   \else
3455     \eq arrange@badness@\m@ne
3456   \fi
3457 }
```

`\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

```
3458 \def\eq arrange@print#1#2{%
3459   \eq restore@hfuzz
3460   \if@eqnsw
3461     \ifdefined\eq tagsleft
3462       \eq tagbox@print@tagsleft
3463     \fi
3464   \fi
3465   \hbox to\displaywidth{%
3466     #1%
3467     \unhbox\eq cellbox@\unkern
3468     #2%
3469     \eq tagging@mathaddlast
3470   }%
3471   \if@eqnsw
3472     \ifdefined\eq tagsleft\else
3473       \eq tagbox@print@tagsright
3474     \fi
3475   \fi
3476 }
```

`ange@print@alignleft` Fit the current equation line with the selected alignment within a given left and right
`ge@print@aligncenter`
`nge@print@alignright`

margins #1 and #2. If we're on the first line, adjust `\eql@display@firstavail@` to the minimum left available space we can guarantee:

```

3477 \def\eql@arrange@print@alignleft#1#2{%
3478   \eql@display@firstavail@set{\dimexpr#1\relax}%
3479   \eql@arrange@print{\kern#1}{\kern#2}%
3480 }

3481 \def\eql@arrange@print@alignright#1#2{%
3482   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3483   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3484 }

3485 \def\eql@arrange@print@aligncenter#1{%
3486   \eql@display@firstavail@set{\dimexpr
3487     (\displaywidth-\eql@cellwidth@+1)/\tw@\relax}%
3488   \ifdim#1>\z@
3489     \eql@arrange@print{\kern#1\hfil}{}%
3490   \else
3491     \eql@arrange@print{\hfil}{\kern-#1}%
3492   \fi
3493 }
```

`\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:

```

3494 \def\eql@arrange@init{%
3495   \eql@save@hfuzz
3496   \hfuzz\maxdimen
3497   \eql@shape@select
3498 }
```

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

```

3499 \def\eql@arrange@print@line{%
3500   \eql@tagging@tagaddbox
3501   \csname eql@arrange%
3502     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3503     @init\endcsname
3504   \csname eql@arrange%
3505     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3506     @\ifdefined\eql@tagpos@reserve
3507       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3508       notag\fi\endcsname
3509 }
```

10.3 Central Alignment

TODO: describe

```

3510 \def\eql@arrange@aligncenter@init{%
3511   \eql@tagging@aligncenter
3512   \eql@line@offset@dimexpr\tw@\eql@shape@amount@
3513     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3514 }
```

TODO: describe

```
3515 \def\eql@arrange@aligncenter@notag{%
3516   \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3517     \ifdim\eql@line@offset@<\eql@marginleft@min@
3518       \dimexpr\tw@eql@marginleft@min@-\eql@line@offset@>\relax
3519     \else
3520       \eql@line@offset@
3521     \fi
3522   \eql@arrange@print@aligncenter\eql@line@offset@
3523 \else
3524   \ifdim\eql@line@offset@<\eql@marginleft@min@
3525     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3526   \else
3527     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3528   \fi
3529 \fi
3530 }
```

TODO: describe

```
3531 \def\eql@arrange@aligncenter@tagsright{%
3532   \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3533     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@>\relax
3534       \dimexpr\tw@eql@marginleft@min@-\eql@line@offset@>\relax
3535     \else
3536       \dimexpr\tw@eql@tagwidth@+\eql@line@offset@>\relax
3537     \fi
3538   \eql@arrange@print@aligncenter\eql@line@offset@
3539 \else
3540   \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@>\relax}%
3541   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3542     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@>\relax
3543       \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3544     \else
3545       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3546     \fi
3547   \else
3548     \let\eql@tagpos@reserve\eql@false
3549     \eql@arrange@aligncenter@notag
3550   \fi
3551 \fi
3552 }
```

```
3553 \def\eql@arrange@aligncenter@tagsleft{%
3554   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3555     \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3556       \ifdim\eql@line@offset@<\eql@tagwidth@
3557         \dimexpr\tw@eql@tagwidth@-\eql@line@offset@>\relax
3558       \else
3559         \eql@line@offset@
3560       \fi
3561     \eql@arrange@print@aligncenter\eql@line@offset@
3562   \else
3563     \eql@arrange@try\eql@tagwidth@
3564     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3565       \ifdim\eql@line@offset@<\eql@tagwidth@
3566         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3567       \else
3568         \eql@arrange@print@alignright\eql@tagwidth@\z@
```

```

3569         \fi
3570     \else
3571         \let\eql@tagpos@reserve\eql@false
3572         \eql@arrange@aligncenter@notag
3573     \fi
3574 \fi
3575 \else
3576     \eql@arrange@aligncenter@notag
3577 \fi
3578 }

```

10.4 Left Alignment

```

3579 \def\eql@arrange@alignleft@init{%
3580     \eql@tagging@alignleft
3581     \eql@line@offset@{\dimexpr\eql@marginleft@+\eql@shape@amount@\relax
3582     \ifdim\eql@line@offset@<\eql@marginleft@min@
3583         \eql@line@offset@{\eql@marginleft@min@
3584     \fi
3585 }

3586 \def\eql@arrange@alignleft@notag{%
3587     \ifdim\eql@line@offset@>\eql@marginleft@min@
3588         \eql@arrange@try\eql@line@offset@
3589         \ifnum\eql@arrange@badness@<\eql@alignbadness@
3590             \eql@arrange@print@alignleft\eql@line@offset@z@
3591         \else
3592             \eql@arrange@print@alignright\eql@marginleft@min@z@
3593         \fi
3594     \else
3595         \eql@arrange@print@alignleft\eql@marginleft@min@z@
3596     \fi
3597 }

3598 \def\eql@arrange@alignleft@tagsright{%
3599     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3600     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3601         \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3602     \else
3603         \ifdim\eql@line@offset@>\eql@marginleft@min@
3604             \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3605         \fi
3606         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3607             \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3608         \else
3609             \let\eql@tagpos@reserve\eql@false
3610             \eql@arrange@alignleft@notag
3611         \fi
3612     \fi
3613 }

3614 \def\eql@arrange@alignleft@tagsleft{%
3615     \ifdim\eql@tagwidth@>\eql@marginleft@min@
3616         \ifdim\eql@line@offset@>\eql@tagwidth@
3617             \eql@arrange@try\eql@line@offset@
3618             \ifnum\eql@arrange@badness@<\eql@alignbadness@
3619                 \eql@arrange@print@alignleft\eql@line@offset@z@
3620             \else
3621                 \eql@arrange@try\eql@tagwidth@
3622             \ifnum\eql@arrange@badness@<\eql@tagbadness@

```

```

3623         \eql@arrange@print@alignright\eql@tagwidth@z@
3624     \else
3625         \let\eql@tagpos@reserve\eql@false
3626         \eql@arrange@print@alignright\eql@marginleft@min@z@
3627     \fi
3628 \fi
3629 \else
3630     \eql@arrange@try\eql@tagwidth@
3631     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3632         \eql@arrange@print@alignleft\eql@tagwidth@z@
3633     \else
3634         \let\eql@tagpos@reserve\eql@false
3635         \eql@arrange@alignleft@notag
3636     \fi
3637 \fi
3638 \else
3639     \eql@arrange@alignleft@notag
3640 \fi
3641 }

```

10.5 Right Alignment

```

3642 \def\eql@arrange@alignright@init{%
3643     \eql@tagging@alignright
3644     \eql@line@offset@\dimexpr\eql@marginright@-\eql@shape@amount@\relax
3645     \ifdim\eql@line@offset@<z@
3646         \eql@line@offset@z@
3647     \fi
3648 }

```

TODO: describe

```

3649 \def\eql@arrange@alignright@notag{%
3650     \ifdim\eql@line@offset@>z@
3651         \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3652         \ifnum\eql@arrange@badness@<\eql@alignbadness@
3653             \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3654         \else
3655             \eql@arrange@print@alignleft\eql@marginleft@min@z@
3656         \fi
3657     \else
3658         \eql@arrange@print@alignright\eql@marginleft@min@z@
3659     \fi
3660 }

```

TODO: describe

```

3661 \def\eql@arrange@alignright@tagsright{%
3662     \ifdim\eql@line@offset@>\eql@tagwidth@
3663         \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3664         \ifnum\eql@arrange@badness@<\eql@alignbadness@
3665             \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3666         \else
3667             \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3668             \ifnum\eql@arrange@badness@<\eql@tagbadness@
3669                 \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3670             \else
3671                 \let\eql@tagpos@reserve\eql@false
3672                 \eql@arrange@print@alignleft\eql@marginleft@min@z@
3673             \fi
3674         \fi

```

```

3675 \else
3676   \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3677   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3678     \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3679   \else
3680     \let\eql@tagpos@reserve\eql@false
3681     \eql@arrange@alignright@notag
3682   \fi
3683 \fi
3684 }

```

TODO: describe

```

3685 \def\eql@arrange@alignright@tagsleft{%
3686   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3687     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3688     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3689       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3690     \else
3691       \ifdim\eql@line@offset@>\z@
3692         \eql@arrange@try\eql@tagwidth@
3693       \fi
3694       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3695         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3696       \else
3697         \let\eql@tagpos@reserve\eql@false
3698         \eql@arrange@alignright@notag
3699       \fi
3700     \fi
3701   \else
3702     \eql@arrange@alignright@notag
3703   \fi
3704 }

```

11 Equations Box Environment

TODO: outline sequence of calls

TODO: describe

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

TODO: vspace?!

11.1 Line Breaks

TODO: describe

`\eql@box@cr`

```

3705 \def\eql@box@cr{%
3706   \ifmmode\else\unskip\fi
3707   \eql@vspaceskip@\z@skip
3708   \let\eql@punct@term\eql@false
3709   \let\eql@class@rel@composed\@empty
3710   \eql@ampprotect\eql@box@cr@test\eql@box@cr@process
3711 }

```

TODO: describe

```

3712 \def\eql@box@cr@test@setopt{\let\eql@box@cr@test\eql@box@cr@testopt}
3713 \def\eql@box@cr@test@setall{\let\eql@box@cr@test\eql@box@cr@testall}

```

\eql@box@cr@testopt **TODO:** describe

```

3714 \def\eql@box@cr@testopt#1{\eql@teststaropt@tight
3715   {\eql@box@cr@testopt@set{#1}}{\eql@box@cr@testopt@set{#1}}{0pt}}
3716 \def\eql@box@cr@testopt@set#1[#2]{\advance\eql@vspaceskip@glueexpr#2\relax#1}

```

\eql@box@cr@testall **TODO:** describe

box@cr@testall@parse

```

3717 \def\eql@box@cr@testall{\eql@parseopt@cr\eql@box@cr@testall@parse}
3718 \def\eql@box@cr@testall@parse{%
3719   \ifx\eql@parseopt@token[%
3720     \let\eql@parseopt@next\eql@parseopt@vspace
3721   \fi
3722   \ifx\eql@parseopt@token*%
3723     \let\eql@parseopt@next\eql@parseopt@gobble
3724   \fi
3725   \ifx\eql@parseopt@token.%
3726     \let\eql@parseopt@next\eql@parseopt@punctpass
3727   \fi
3728   \ifx\eql@parseopt@token,%
3729     \let\eql@parseopt@next\eql@parseopt@punctpass
3730   \fi
3731   \ifx\eql@parseopt@token~%
3732     \let\eql@parseopt@next\eql@parseopt@punctpass
3733   \fi
3734   \ifx\eql@parseopt@token'%
3735     \let\eql@parseopt@next\eql@parseopt@punctnext
3736   \fi
3737   \ifx\eql@parseopt@token!%
3738     \let\eql@parseopt@next\eql@parseopt@punctterm
3739   \fi
3740   \ifx\eql@parseopt@token/%
3741     \let\eql@parseopt@next\eql@parseopt@punctclear
3742   \fi
3743   \ifx\eql@parseopt@token=%
3744     \let\eql@parseopt@next\eql@parseopt@relsymp
3745   \fi
3746   \ifx\eql@parseopt@token;%
3747     \let\eql@parseopt@next\eql@parseopt@relcont
3748   \fi
3749   \ifx\eql@parseopt@token:%
3750     \let\eql@parseopt@next\eql@parseopt@relstart
3751   \fi
3752   \ifx\eql@parseopt@token|%
3753     \let\eql@parseopt@next\eql@parseopt@relord
3754   \fi
3755   \ifx\eql@parseopt@token?%
3756     \let\eql@parseopt@next\eql@box@cr@parse@rel
3757   \fi
3758   \ifx\eql@parseopt@token&%
3759     \let\eql@parseopt@next\eql@parseopt@end
3760   \fi
3761 }
3762 \def\eql@box@cr@parse@rel#1#2{%
3763   \def\eql@tmp{#2}%
3764   \ifx\eql@tmp\eql@relax\else

```

```

3765 \ifdefined\eql@punct@next\else
3766 \eql@punct@next@clear
3767 \fi
3768 \fi
3769 \ifx\eql@box@open\eql@box@lines@open
3770 \ifx\eql@tmp@empty
3771 \def\eql@class@rel@composed{\eql@shape@cont}%
3772 \else
3773 \def\eql@class@rel@composed{\eql@shape@rel#2}%
3774 \fi
3775 \fi
3776 \ifx\eql@box@open\eql@box@columns@open
3777 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3778 \fi
3779 \ifx\eql@box@open\eql@box@cases@open
3780 \def\eql@class@rel@composed{&#2}%
3781 \fi
3782 \eql@parseopt@end}

```

`\eql@box@cr@process`

```

3783 \def\eql@box@cr@process{%
3784 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
3785 \edef\eql@tmp{%
3786 \unexpanded{%
3787 \eql@box@endline
3788 \eql@box@lastcell
3789 \cr
3790 }%
3791 \noalign{%
3792 \vskip\the\eql@vspaceskip@relax
3793 }%
3794 \unexpanded\expandafter{\eql@class@rel@composed}%
3795 }%
3796 \eql@tmp
3797 }

```

`\eql@box@endline`

```

3798 \def\eql@box@endline{%
3799 \eql@punct@apply@line
3800 \eql@hook@lineout
3801 }

```

11.2 Column Breaks

TODO: describe

```

3802 \def\eql@box@amp{%
3803 \eql@ampprotecttwo\eql@box@amp@testescape\eql@amp@org
3804 \eql@box@amp@process}
3805 \def\eql@box@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
3806 \relax
3807 \let\eql@punct@term\eql@false
3808 \let\eql@class@rel@composed\empty
3809 \eql@box@amp@test{#2}}}

```

TODO: describe


```

3810 \def\eql@box@amp@test@setopt{%
3811   \let\eql@box@amp@test\eql@box@amp@testopt}
3812 \def\eql@box@amp@test@setall{%
3813   \let\eql@box@amp@test\eql@box@amp@testall}

\eql@box@amp@testopt TODO: describe

3814 \let\eql@box@amp@testopt\@empty

\eql@box@amp@testall TODO: describe
ox@amp@testall@parse
3815 \def\eql@box@amp@testall{\eql@parseopt@cr\eql@box@amp@testall@parse}
3816 \def\eql@box@amp@testall@parse{%
3817   \ifx\eql@parseopt@token.%
3818     \let\eql@parseopt@next\eql@parseopt@punctpass
3819   \fi
3820   \ifx\eql@parseopt@token,%
3821     \let\eql@parseopt@next\eql@parseopt@punctpass
3822   \fi
3823   \ifx\eql@parseopt@token~%
3824     \let\eql@parseopt@next\eql@parseopt@punctpass
3825   \fi
3826   \ifx\eql@parseopt@token'%
3827     \let\eql@parseopt@next\eql@parseopt@punctnext
3828   \fi
3829   \ifx\eql@parseopt@token!%
3830     \let\eql@parseopt@next\eql@parseopt@punctterm
3831   \fi
3832   \ifx\eql@parseopt@token=%
3833     \let\eql@parseopt@next\eql@parseopt@relsymp
3834   \fi
3835   \ifx\eql@parseopt@token;%
3836     \let\eql@parseopt@next\eql@parseopt@relcont
3837   \fi
3838   \ifx\eql@parseopt@token:%
3839     \let\eql@parseopt@next\eql@parseopt@relstart
3840   \fi
3841   \ifx\eql@parseopt@token|%
3842     \let\eql@parseopt@next\eql@parseopt@relord
3843   \fi
3844   \ifx\eql@parseopt@token?%
3845     \let\eql@parseopt@next\eql@box@amp@parse@rel
3846   \fi
3847   \ifx\eql@parseopt@token&%
3848     \let\eql@parseopt@next\eql@parseopt@end
3849   \fi
3850 }
3851 \def\eql@box@amp@parse@rel#1#2{%
3852   \ifx\eql@box@open\eql@box@columns@open
3853     \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3854   \else
3855     \def\eql@class@rel@composed{#2}%
3856   \fi
3857   \eql@parseopt@end}

\eql@box@amp@process

3858 \def\eql@box@amp@process{%
3859   \ifdefined\eql@punct@term\eql@punct@apply@top\fi

```

```

3860 \ifx\eql@box@open\eql@box@columns@open
3861 \edef\eql@tmp{%
3862 \ifx\eql@class@rel@composed\@empty
3863 \ifx\eql@box@lastcell\eql@box@columns@lastcell@odd
3864 &\noexpand\eql@punct@next@clear\fi&%
3865 \else
3866 \ifx\eql@box@lastcell\eql@box@columns@lastcell@even&\fi%
3867 \unexpanded\expandafter{\eql@class@rel@composed}%
3868 \fi
3869 }%
3870 \else
3871 \edef\eql@tmp{%
3872 \ifx\eql@class@rel@composed\@empty
3873 &%
3874 \else
3875 \unexpanded\expandafter{\eql@class@rel@composed}%
3876 \fi
3877 }%
3878 \fi
3879 \eql@tmp
3880 }

```

11.3 Lines Mode

```

3881 \def\eql@box@lines@lastcell{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3882 \def\eql@box@lines@open{%
3883 \eql@shape@align@enable
3884 \let\eql@box@lastcell\eql@box@lines@lastcell
3885 \everycr{\noalign{%
3886 \eql@verbose@info\eql@verbose@msg@startline@number
3887 \global\advance\eql@row@\@ne
3888 }}%
3889 \tabskip\z@skip
3890 \halign\bgroup
3891 &%
3892 \global\let\eql@cell@container\@empty
3893 \setbox\eql@cellbox@\hbox{%
3894 \eql@strut@cell
3895 \@lign
3896 $\m@th\eql@mathstyle
3897 \eql@hook@colin
3898 ##%
3899 \eql@punct@apply@col
3900 \eql@hook@colout
3901 \eql@tagging@mathsave
3902 $%
3903 \eql@tagging@mathaddlast
3904 }%
3905 \ifdefined\eql@shape@lastrow
3906 \eql@totalrows@\eql@row@
3907 \fi
3908 \eql@shape@eval
3909 \eql@cell@container
3910 \ifdefined\eql@frame@cmd
3911 \ifcase\eql@shape@pos@
3912 \eql@frame@measure

```

```

3913         \advance\eq@shape@amount@-\eq@frame@margin@
3914     \or\or
3915         \eq@frame@measure
3916         \advance\eq@shape@amount@+\eq@frame@margin@
3917     \fi
3918     \eq@frame@print
3919 \fi
3920 \ifcase\eq@shape@pos@
3921     \kern\eq@shape@amount@
3922     \box\eq@cellbox@
3923     \hskip\glueexpr\eq@paddingleft@+\eq@paddingright@
3924     -\eq@shape@amount@+\@flushglue\relax
3925     \eq@tagging@alignleft
3926 \or
3927     \hskip\glueexpr\eq@paddingleft@+\eq@shape@amount@+\@flushglue\relax
3928     \box\eq@cellbox@
3929     \hskip\glueexpr\eq@paddingright@-\eq@shape@amount@+\@flushglue\relax
3930     \eq@tagging@aligncenter
3931 \or
3932     \hskip\glueexpr\eq@paddingleft@+\eq@paddingright@
3933     +\eq@shape@amount@+\@flushglue\relax
3934     \box\eq@cellbox@
3935     \kern-\eq@shape@amount@
3936     \eq@tagging@alignright
3937 \fi
3938     \tabskip\eq@colsep@\relax
3939 \crrc
3940 \noalign{%
3941     \global\let\eq@shape@lastrow\eq@false
3942     \eq@hook@blockbefore
3943 }%
3944 \eq@hook@blockin
3945 }
3946 \def\eq@box@lines@set{\let\eq@box@open\eq@box@lines@open}

```

11.4 Columns Mode

```

3947 \def\eq@box@columns@lastcell@odd{%
3948     &\omit
3949     \eq@prevwidth@\wd\eq@cellbox@
3950     \let\eq@frame@cmd\eq@frame@prevcmd
3951     \ifdefined\eq@frame@cmd
3952         \eq@frame@measure
3953         \advance\eq@prevwidth@\eq@frame@margin@
3954         \eq@frame@print
3955     \fi
3956     \kern-\eq@prevwidth@
3957     \unhbox\eq@cellbox@
3958     \hfil
3959     &\omit\kern-\eq@colsep@
3960 }%
3961 \def\eq@box@columns@lastcell@even{&\omit\kern-\eq@colsep@}
3962 \def\eq@box@columns@open{%
3963 % \TODO templates
3964     \eq@shape@align@disable
3965     \let\eq@box@lastcell@empty
3966     \everycr{\noalign{%
3967         \eq@verbose@info\eq@verbose@msg@startline@new

```

```

3968 }}%
3969 \tabskip\z@skip
3970 \halign\bgroup
3971   &%
3972   \let\eql@box@lastcell\eql@box@columns@lastcell@odd
3973   \global\let\eql@cell@container\@empty
3974   \global\setbox\eql@cellbox@\hbox{%
3975     \eql@strut@cell
3976     \@lign
3977     $\m@th\eql@mathstyle
3978     \eql@hook@colin
3979     ##%
3980     \eql@punct@apply@next
3981     \eql@class@innerleft
3982     \eql@hook@innerleft
3983     \eql@tagging@mathsave
3984     $%
3985     \eql@tagging@mathaddlast
3986   }%
3987   \eql@cell@container
3988   \hfil
3989   \kern\wd\eql@cellbox@
3990   \ifdefined\eql@frame@cmd
3991     \eql@frame@measure
3992     \kern\eql@frame@margin@
3993   \fi
3994   \global\let\eql@frame@prevcmd\eql@frame@cmd
3995   \tabskip\z@skip
3996   &%
3997   \eql@prevwidth@\wd\eql@cellbox@
3998   \let\eql@box@lastcell\eql@box@columns@lastcell@even
3999   \let\eql@frame@cmd\eql@frame@prevcmd
4000   \global\let\eql@cell@container\@empty
4001   \setbox\eql@cellbox@\hbox{%
4002     \unhbox\eql@cellbox@
4003     \eql@strut@cell
4004     \@lign
4005     $\m@th\eql@mathstyle
4006     \eql@hook@innerright
4007     \eql@class@innerright@sel
4008     ##%
4009     \eql@punct@apply@col
4010     \eql@hook@colout
4011     \eql@tagging@mathsave
4012     $%
4013     \eql@tagging@mathaddlast
4014   }%
4015   \eql@cell@container
4016   \ifdefined\eql@frame@cmd
4017     \eql@frame@measure
4018     \advance\eql@prevwidth@\eql@frame@margin@
4019     \eql@frame@print
4020   \fi
4021   \kern-\eql@prevwidth@
4022   \unhbox\eql@cellbox@
4023   \hfil
4024   \tabskip\eql@colsep@\relax
4025 \crcr

```

```

4026 \noalign{%
4027 \eq@hook@blockbefore
4028 }%
4029 \eq@hook@blockin
4030 }

4031 \def\eq@box@columns@set{\let\eq@box@open\eq@box@columns@open}

```

11.5 Cases Mode

TODO: describe

TODO: how to get proper height in tagging (and avoid nulldelimiterspace) **TODO:** add alignment?

```

4032 \def\eq@box@cases@lastcell{&}%

4033 \let\eq@box@cases@condtext\eq@false
4034 \let\eq@box@cases@condintro\@empty

4035 \def\eq@box@cases@open{%
4036 \eq@shape@align@disable
4037 \let\eq@box@lastcell\@empty
4038 \everycr{\noalign{%
4039 \eq@verbose@info\eq@verbose@msg@startline@new
4040 }}%
4041 \tabskip\z@skip
4042 \halign\bgroup
4043 \let\eq@box@lastcell\eq@box@cases@lastcell
4044 \global\let\eq@cell@container\@empty
4045 \global\setbox\eq@cellbox@\hbox{%
4046 \eq@strut@cell
4047 \@lign
4048 $\m@th\eq@mathstyle
4049 \eq@hook@colin
4050 ##%
4051 \eq@punct@apply@col
4052 \eq@tagging@mathsave
4053 $%
4054 \eq@tagging@mathaddlast
4055 }%
4056 \eq@cell@container
4057 \unhbox\eq@cellbox@
4058 \hfil
4059 \eq@tagging@alignleft
4060 \tabskip\eq@colsep@\relax
4061 &%
4062 \let\eq@box@lastcell\@empty
4063 \global\let\eq@cell@container\@empty
4064 \setbox\eq@cellbox@\hbox{%
4065 \unhbox\eq@cellbox@
4066 \eq@strut@cell
4067 \@lign
4068 $\m@th\eq@mathstyle
4069 \ifdefined\eq@box@cases@condtext
4070 \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
4071 \eq@box@cases@condintro
4072 ##%
4073 \eq@punct@apply@col
4074 \egroup

```

```

4075         \eql@hook@colout
4076         \eql@tagging@mathsave
4077         $%
4078         \eql@tagging@mathaddlast
4079     }%
4080     \eql@cell@container
4081     \unhbox\eql@cellbox@
4082     \hfil
4083     \eql@tagging@alignleft
4084     \tabskip\z@skip
4085     \crrc
4086     \noalign{%
4087         \eql@hook@blockbefore
4088     }%
4089     \eql@hook@blockin
4090 }

4091 \def\eql@box@cases@set{%
4092     \ifdefined\eql@cases@mathstyle\let\eql@mathstyle\eql@cases@mathstyle\fi
4093     \ifdefined\eql@punct@cases\let\eql@punct@col\eql@punct@cases\fi
4094     \let\eql@box@open\eql@box@cases@open}

```

11.6 Main

```

4095 \let\eql@box@box\center
4096 \let\eql@box@open\@undefined
4097 \let\eql@box@frame\@firstofone
4098 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

4099 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4100     \left#1##1\right#2}}
4101 \def\eql@box@getdim{\setbox\@ne\hbox{\@ne\ht\@ne\ht\z@\dp\@ne\dp\z@}
4102     \def\eql@box@deldim#1{\hbox{$\m@th\hbox{\null\delimiterspace\z@\left#1%
4103         \ifx\eql@box@box\center\center{\box\@ne}\else\box\@ne\fi\right.$}}
4104     \def\eql@box@ldelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4105         \eql@box@getdim\eql@box@deldim#1##1}}
4106     \def\eql@box@rdelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4107         \eql@box@getdim#1\eql@box@deldim#1}}

```

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

```

4108 \def\eql@box@close{%
4109     \ifvmode\else
4110         \ifmmode\else\unskip\fi
4111         \global\let\eql@shape@lastrow\eql@true
4112         \eql@punct@apply@block
4113         \ifdefined\eql@box@punct@term
4114             \eql@punct@apply@top
4115         \fi
4116         \eql@box@endline
4117         \eql@box@lastcell
4118         \cr
4119     \fi
4120     \noalign{%
4121         \eql@hook@blockafter

```

```

4122     \global\let\eq@shape@lastrow\eq@false
4123   }%
4124   \eq@tagging@tablesaveinner
4125 \egroup
4126 }

```

\eq@box@vcenter

```

4127 \def\eq@box@vcenter#1{%
4128   \ifmmode
4129     \vcenter{#1}%
4130   \else
4131     $\m@th\vcenter{#1}$%
4132   \fi
4133 }

```

\eq@box@start

```

4134 \let\eq@box@endmath\eq@false
4135 \def\eq@box@start{%
4136   \relax
4137   \ifmmode
4138     \let\eq@box@endmath\eq@false
4139   \else
4140     \let\eq@box@endmath\eq@true
4141     \expandafter$%$
4142   \fi
4143   \eq@stack@save@box
4144   \let\eq@frame@cmd\@undefined
4145   \let\eq@layoutleft\eq@false
4146   \eq@row@z@
4147   \eq@totalrows@\@M
4148   \eq@shape@select
4149   \setbox\z@\ifx\eq@box@box\vcenter
4150     \expandafter\vbox
4151   \else
4152     \expandafter\eq@box@box
4153   \fi\bgroup
4154   \let\eqnpunct\eq@punct@setnext
4155   \eq@display@nest
4156   \let\\\eq@box@cr
4157   \ifdefined\eq@amp@mode
4158     \let\&\eq@box@amp
4159   \fi
4160   \eq@spread@set
4161   \eq@strut@make
4162   \eq@box@open
4163 }

```

\eq@box@end

```

4164 \def\eq@box@end{%
4165   \eq@box@close
4166 \egroup
4167 \eq@box@frame{%
4168   \ifdefined\eq@display@marginleft
4169     \hskip\glueexpr\eq@display@marginleft\relax
4170   \fi
4171   \ifx\eq@box@box\vcenter

```

```

4172     \eql@box@vcenter{\unvbox\z@}%
4173   \else
4174     \box\z@
4175   \fi
4176   \eql@tagging@tableaddinner
4177   \ifdefined\eql@display@marginright
4178     \hskip\glueexpr\eql@display@marginright\relax
4179   \fi
4180 }%
4181 \eql@stack@restore
4182 \ifdefined\eql@box@endmath
4183   \expandafter$%$
4184 \fi
4185 }

```

`\eql@box@main` Combined opening, body and closing for pre-scanned body:

```

4186 \def\eql@box@main{%
4187   \eql@box@start
4188   \eql@scan@body
4189   \eql@box@end
4190 }

```

11.7 Options Processing

TODO: describe

```

4191 \def\eql@box@test@setopt{\let\eql@box@test\eql@box@testopt}
4192 \def\eql@box@test@setall{\let\eql@box@test\eql@box@testall}

```

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

```

4193 \def\eql@box@testopt#1{\eql@testopt@tight{\eql@box@testopt@set{#1}}{}}
4194 \def\eql@box@testopt@set#1[#2]{\eqnaddopt{#2}#1}

4195 \def\eql@box@testall{\eql@parseopt@env\eql@box@testall@parse}
4196 \def\eql@box@testall@parse{%
4197   \ifx\eql@parseopt@token[%
4198     \let\eql@parseopt@next\eql@parseopt@opt
4199   \fi
4200   \ifx\eql@parseopt@token.%
4201     \let\eql@parseopt@next\eql@parseopt@punctpass
4202   \fi
4203   \ifx\eql@parseopt@token,%
4204     \let\eql@parseopt@next\eql@parseopt@punctpass
4205   \fi
4206   \ifx\eql@parseopt@token~%
4207     \let\eql@parseopt@next\eql@parseopt@punctpass
4208   \fi
4209   \ifx\eql@parseopt@token'
4210     \let\eql@parseopt@next\eql@parseopt@punctopt
4211   \fi
4212   \ifx\eql@parseopt@token!%
4213     \let\eql@parseopt@next\eql@box@parse@punctterm
4214   \fi
4215   \ifx\eql@parseopt@token=%
4216     \let\eql@parseopt@next\eql@parseopt@lines
4217   \fi

```



```

4218 \ifx\eql@parseopt@token|%
4219 \let\eql@parseopt@next\eql@parseopt@columns
4220 \fi
4221 \ifx\eql@parseopt@token<%
4222 \let\eql@parseopt@next\eql@parseopt@ampeq
4223 \fi
4224 \ifx\eql@parseopt@token>%
4225 \let\eql@parseopt@next\eql@parseopt@eqamp
4226 \fi
4227 }
4228 \def\eql@box@parse@punctterm#1{\eqnaddopt{punctterm}\eql@parseopt@peek}

```

\eql@box@end@testall **TODO:** describe

```

4229 \def\eql@box@end@testall{\eql@parseopt@env\eql@box@end@testall@parse}
4230 \def\eql@box@end@testall@parse{%
4231 \ifx\eql@parseopt@token.%
4232 \let\eql@parseopt@next\eql@parseopt@punctpass
4233 \fi
4234 \ifx\eql@parseopt@token,%
4235 \let\eql@parseopt@next\eql@parseopt@punctpass
4236 \fi
4237 \ifx\eql@parseopt@token~%
4238 \let\eql@parseopt@next\eql@parseopt@punctpass
4239 \fi
4240 \ifx\eql@parseopt@token'%
4241 \let\eql@parseopt@next\eql@parseopt@punctblock
4242 \fi
4243 \ifx\eql@parseopt@token!%
4244 \let\eql@parseopt@next\eql@box@end@parse@punctterm
4245 \fi
4246 }
4247 \def\eql@box@end@parse@punctterm#1{%
4248 \let\eql@box@punct@term\eql@true\eql@parseopt@peek}

```

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

```

4249 \def\eql@box@processopt{%
4250 \let\eql@box@frame\@firstofone
4251 \let\eql@display@marginleft\@undefined
4252 \let\eql@display@marginright\@undefined
4253 \let\eql@box@punct@term\eql@false
4254 \let\eql@punct@block\@undefined
4255 \eql@nextopt@process{box}%
4256 \let\eql@punct@next\@undefined
4257 \eql@colsep@\glueexpr\eql@box@colsep\relax
4258 \ifdefined\eql@paddingleft@val
4259 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
4260 \else
4261 \eql@paddingleft@\z@
4262 \fi
4263 \ifdefined\eql@paddingright@val
4264 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
4265 \else
4266 \eql@paddingright@\z@
4267 \fi
4268 \eql@indent@\glueexpr\eql@indent@val\relax
4269 }

```

11.8 Environment

`equationsbox` (*env.*)

```

4270 \newenvironment{equationsbox}{%
4271   \eql@verbose@info\eql@verbose@msg@enterenv
4272   \ifdefined\eql@box@env@modifier
4273     \eql@box@test@setall
4274   \else
4275     \eql@box@nomodifier
4276   \fi
4277   \eql@ampprotect\eql@box@test\eql@box@env@start
4278 }{%
4279   \ifdefined\eql@box@doscan\else
4280     \expandafter\eql@box@end
4281   \fi
4282   \eql@verbose@info\eql@verbose@msg@leaveenv
4283 }
```

`\eql@box@env@start`

```

4284 \def\eql@box@env@start{%
4285   \eql@box@processopt
4286   \ifdefined\eql@box@doscan
4287     \eql@box@call@set
4288     \expandafter\eql@scan@env
4289   \else
4290     \expandafter\eql@box@start
4291   \fi
4292 }
```

`\eql@box@call`

```

4293 \def\eql@box@call{\eql@box@main\eql@scan@end}
4294 \def\eql@box@call@test{%
4295   \eql@ampprotect\eql@box@end@testall\eql@box@call}
4296 \def\eql@box@call@set{%
4297   \ifdefined\eql@box@end@modifier
4298     \let\eql@scan@call\eql@box@call@test
4299   \else
4300     \let\eql@scan@call\eql@box@call
4301   \fi
4302 }
```

`\eql@box@ang@open`

```

4303 \newenvironment{equationsbox@ang}{-}{-}
4304 \def\eql@box@ang@open{%
4305   \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4306   \begin{equationsbox@ang}%
4307   \eql@verbose@info\eql@verbose@msg@enterenv
4308   \let\>\eql@box@ang@close
4309   \ifdefined\eql@box@ang@modifier
4310     \eql@box@test@setall
4311   \else
4312     \eql@box@nomodifier
4313   \fi
4314   \eql@ampprotect\eql@box@test\eql@box@ang@start
4315 }
```

`\eql@box@ang@start` Process arguments and start handling the box:

```
4316 \def\eql@box@ang@start{%
4317   \eql@box@processopt
4318   \ifdefined\eql@box@doscan
4319     \eql@box@call@set
4320     \expandafter\eql@scan@ang
4321   \else
4322     \expandafter\eql@box@start
4323   \fi
4324 }
```

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

```
4325 \def\eql@box@ang@close{%
4326   \ifdefined\eql@box@doscan
4327     \let\eql@box@end@modifier\eql@false
4328   \fi
4329   \ifdefined\eql@box@end@modifier
4330     \expandafter\eql@ampprotect\expandafter\eql@box@end@testall
4331   \fi
4332   \eql@box@ang@end
4333 }
```

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

```
4334 \def\eql@box@ang@end{%
4335   \ifdefined\eql@box@doscan\else
4336     \expandafter\eql@box@end
4337   \fi
4338   \eql@verbose@info\eql@verbose@msg@leaveenv
4339   \end{equationsbox@ang}%
4340   \ignorespaces
4341 }
```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```
4342 \def\eql@single@start@native{%
4343   \eql@display@init
4344   \eql@display@print
4345   \let\raisetag\eql@raisetag@default
4346   \eql@shape@align@disable
4347   \eql@hook@eqin
4348 }%
```

TODO: describe

```
4349 \def\eql@single@end@native{%
4350   \eql@tags@container
4351   \eql@numbering@single@eval
4352   \if@eqnsw
4353     \ifdefined\eql@tagsleft
4354       \leqno
```

```

4355     \else
4356     \eqno
4357     \fi
4358     \eql@composetag@print
4359 \fi
4360 \eql@interline@container
4361 \advance\eql@belowspace@\eql@vspaceskip@
4362 \eql@display@container
4363 \eql@display@penalty
4364 \eql@display@vspace@native
4365 }%

```

12.2 Print

```

4366 \def\eql@single@start@print{%
4367   \eql@display@init
4368   \eql@display@print
4369   \eql@shape@align@enable

4370   \eql@totalrows@\@ne
4371   \eql@row@\@ne
4372   \eql@arrange@init
4373   \global\let\eql@cell@container\@empty

4374   \prevgraf\numexpr\prevgraf+\@ne\relax
4375   \setbox\eql@cellbox@\hbox\bgroup
4376     \eql@restore@hfuzz
4377     \eql@strut@cell
4378     $\m@th\eql@mathstyle%$
4379     \eql@hook@eqin
4380 }

4381 \def\eql@single@end@print{%
4382   \eql@tagging@mathsave
4383   $%$
4384   \hfil
4385   \kern\z@
4386 \egroup
4387 \prevgraf\numexpr\prevgraf-\@ne\relax

4388 \eql@shape@eval
4389 \eql@cell@container

4390 \ifdefined\eql@frame@cmd
4391   \eql@frame@adjust
4392 \fi

4393 \eql@cellwidth@\wd\eql@cellbox@
4394 \eql@line@height@\ht\eql@cellbox@
4395 \eql@line@depth@\dp\eql@cellbox@
4396 \eql@totalwidth@\eql@cellwidth@
4397 \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
4398 \eql@topheight@\eql@line@height@
4399 \eql@bottomdepth@\eql@line@depth@

4400 \eql@tags@container
4401 \eql@numbering@single@eval
4402 \if@eqnsw
4403   \eql@tagbox@make\eql@composetag@print
4404   \eql@tagrows@\@ne
4405   \ifdefined\eql@tagpos@reserve\else
4406     \eql@tagwidth@\z@
4407   \fi

```

```

4408 \eql@tagheight@block@\ht\eql@tagbox@
4409 \eql@tagdepth@block@\dp\eql@tagbox@
4410 \else
4411 \eql@numbering@warnunused
4412 \eql@tagwidth@\z@
4413 \eql@taggrows@\z@
4414 \fi
4415 \eql@tagwidth@max@\eql@tagwidth@
4416 \eql@tagpos@single@eval
4417 \eql@tagpos@print@line@eval
4418 \eql@intercolumns@\z@
4419 \eql@adjust@calc@lines
4420 \eql@display@halign@init{}%
4421 \halign{##\crr
4422 \noalign{\eql@display@halign@start}%
4423 \eql@arrange@print@line
4424 \cr
4425 \noalign{\eql@display@halign@end}%
4426 \eql@tagging@tablesavelines
4427 }%
4428 \eql@tagpos@print@line@end
4429 \eql@display@close
4430 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```

4431 \def\eql@lines@measure@line@begin{%
4432 \eql@verbose@info\eql@verbose@msg@startline@number
4433 \eql@numbering@measure@line@begin
4434 \eql@hook@linein
4435 }

```

TODO: describe

```

4436 \def\eql@lines@measure@line@end{%
4437 \eql@punct@apply@line
4438 \eql@hook@lineout
4439 }

```

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

```

4440 \def\eql@lines@measure@cell{%
4441 \ifdefined\eql@frame@cmd
4442 \ifcase\eql@shape@pos@
4443 \eql@frame@measure
4444 \advance\eql@shape@amount@-\eql@frame@margin@
4445 \or\or
4446 \eql@frame@measure
4447 \advance\eql@shape@amount@+\eql@frame@margin@
4448 \fi

```

```

4449 \eql@frame@print
4450 \fi
4451 \eql@cellwidth@\wd\eql@cellbox@
4452 \eql@line@height@\ht\eql@cellbox@
4453 \eql@line@depth@\dp\eql@cellbox@
4454 \eql@dimensions@startrow
4455 \eql@dimensions@savecell
4456 \kern\eql@cellwidth@
4457 }

```

\eql@lines@measure

```

4458 \def\eql@lines@measure{%
4459 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
4460 \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
4461 \ifdefined\eql@amp@mode
4462 \let\&\eql@break@amp
4463 \fi
4464 \eql@totalrows@\@M
4465 \eql@shape@select

4466 \setbox\z@\vbox{\measuring@true\halign{%
4467 \global\let\eql@cell@container\@empty
4468 \setbox\eql@cellbox@\hbox{%
4469 \eql@strut@cell
4470 \@lign
4471 $\m@th\eql@mathstyle
4472 \eql@hook@colin
4473 ##%
4474 \eql@punct@apply@col
4475 \eql@hook@colout
4476 $%
4477 }%
4478 \ifdefined\eql@shape@lastrow
4479 \eql@totalrows@\eql@row@
4480 \fi
4481 \eql@shape@eval
4482 \eql@cell@container
4483 \eql@lines@measure@cell
4484 \eql@measure@tag
4485 \eql@measure@endrow
4486 \crcr

4487 \noalign{%
4488 \global\let\eql@shape@lastrow\eql@false
4489 \eql@hook@blockbefore
4490 }%
4491 \eql@hook@blockin
4492 \eql@scan@body
4493 \ifvmode\else
4494 \global\let\eql@shape@lastrow\eql@true
4495 \eql@punct@apply@block
4496 \eql@hook@blockout
4497 \eql@multi@endline
4498 \cr
4499 \fi
4500 \omit
4501 \cr
4502 \noalign{%

```

```

4503     \eql@hook@blockafter
4504     \global\let\eql@shape@lastrow\eql@false
4505 }%
4506 }}%

4507 \eql@measure@close

4508 \setbox\z@\vbox{%
4509     \unvbox\z@
4510     \unpenalty
4511     \global\setbox\@ne\lastbox
4512 }%
4513 \eql@totalwidth@\wd\@ne

4514 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@measure
4515 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

4516 \def\eql@lines@adjust{%
4517     \eql@tagpos@adjust@eval
4518     \eql@adjust@calc@lines
4519     \eql@numbering@best@eval
4520 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```

4521 \def\eql@lines@print@line@begin{%
4522     \eql@verbose@info\eql@verbose@msg@startline@number
4523     \eql@numbering@print@line@begin
4524     \eql@hook@linein
4525 }

```

TODO: describe

```

4526 \def\eql@lines@print@line@end{%
4527     \eql@punct@apply@line
4528     \eql@hook@lineout
4529 }

```

TODO: describe

```

4530 \def\eql@lines@print@line@adjust{%
4531     \ifdefined\eql@frame@cmd
4532     \ifcase\eql@shape@pos@
4533         \eql@frame@measure
4534         \advance\eql@shape@amount@-\eql@frame@margin@
4535     \or\or
4536         \eql@frame@measure
4537         \advance\eql@shape@amount@+\eql@frame@margin@
4538     \fi
4539     \eql@frame@adjust
4540 \fi

```

```

4541 \eql@cellwidth@\wd\eql@cellbox@
4542 \eql@line@height@\ht\eql@cellbox@
4543 \eql@line@depth@\dp\eql@cellbox@
4544 \eql@numbering@print@line@eval
4545 \if@eqnsw
4546   \eql@tagbox@make\eql@composetag@print
4547 \fi
4548 \eql@tagpos@print@line@eval
4549 \eql@arrange@print@line
4550 \eql@tagpos@print@line@end
4551 }

```

TODO: describe

```

4552 \def\eql@lines@print{%
4553   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@print
4554   \eql@arrange@init
4555   \eql@display@halign@init\eql@lines@print@line@begin
4556   \eql@multi@cr@let\eql@lines@print@line@end
4557   \ifdefined\eql@amp@mode
4558     \let\&\eql@break@amp
4559   \fi
4560   \tabskip\z@skip

4561   \halign{%
4562     \global\let\eql@cell@container\@empty
4563     \setbox\eql@cellbox@\hbox{%
4564       \eql@restore@hfuzz
4565       \eql@strut@cell
4566       \@lign
4567       $\m@th\eql@mathstyle
4568       \eql@hook@colin
4569       ##%
4570       \eql@punct@apply@col
4571       \eql@hook@colout
4572       \eql@tagging@mathsave
4573       $%
4574       \hfil
4575       \kern\z@
4576     }%
4577     \eql@shape@eval
4578     \eql@cell@container
4579     \eql@lines@print@line@adjust
4580   \crr

4581   \noalign{%
4582     \eql@display@halign@start
4583     \eql@numbering@print@block@begin
4584     \eql@hook@blockbefore
4585   }%
4586   \eql@hook@blockin
4587   \eql@scan@body
4588   \ifvmode\else
4589     \relax
4590     \eql@punct@apply@block
4591     \eql@hook@blockout
4592     \eql@multi@endline
4593   \cr
4594   \fi
4595   \noalign{%

```



```

4596     \eql@hook@blockafter
4597     \eql@display@halign@end
4598     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@print
4599 }%
4600 \eql@tagging@tablesavelines
4601 }%
4602 }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```

\eql@columns@add@amp
@columns@completerow
4603 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
4604 \def\eql@columns@completerow{%
4605   \count@\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
4606   \edef\eql@tmp{%
4607     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
4608   \eql@tmp
4609 }

4610 \def\eql@columns@overfull{%
4611   \dimen@\eql@line@width@
4612   \advance\dimen@-\hfuzz
4613   \ifdim\dimen@>\displaywidth
4614     \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
4615     \wd\z@\z@
4616     \ht\z@\eql@line@height@
4617     \dp\z@\eql@line@depth@
4618     \box\z@
4619   \fi
4620 }

```

14.2 Column Breaks

TODO: describe

TODO: describe

```

4621 \let\eql@amp@org&

4622 \def\eql@columns@amp{%
4623   \eql@ampprotecttwo\eql@columns@amp@testescape\eql@amp@org
4624   \eql@columns@amp@process}
4625 \def\eql@columns@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
4626   \relax
4627   \let\eql@punct@term\eql@false
4628   \let\eql@class@rel@composed\@empty
4629   \eql@columns@amp@test{#2}}}

```

TODO: describe

```

4630 \def\eql@columns@amp@test@setopt{%
4631   \let\eql@columns@amp@test\eql@columns@amp@testopt}
4632 \def\eql@columns@amp@test@setall{%
4633   \let\eql@columns@amp@test\eql@columns@amp@testall}

```

@columns@amp@testopt **TODO:** describe

```

4634 \let\eql@columns@amp@testopt\@empty

```

@columns@amp@testall **TODO:** describe

ns@amp@testall@parse

```

4635 \def\eql@columns@amp@testall{\eql@parseopt@cr\eql@columns@amp@testall@parse}
4636 \def\eql@columns@amp@testall@parse{%
4637   \ifx\eql@parseopt@token.%
4638     \let\eql@parseopt@next\eql@parseopt@punctpass
4639   \fi
4640   \ifx\eql@parseopt@token,%
4641     \let\eql@parseopt@next\eql@parseopt@punctpass
4642   \fi
4643   \ifx\eql@parseopt@token~%
4644     \let\eql@parseopt@next\eql@parseopt@punctpass
4645   \fi
4646   \ifx\eql@parseopt@token'%
4647     \let\eql@parseopt@next\eql@parseopt@punctnext
4648   \fi
4649   \ifx\eql@parseopt@token!%
4650     \let\eql@parseopt@next\eql@parseopt@punctterm
4651   \fi
4652   \ifx\eql@parseopt@token=%
4653     \let\eql@parseopt@next\eql@parseopt@relsymp
4654   \fi
4655   \ifx\eql@parseopt@token;%
4656     \let\eql@parseopt@next\eql@parseopt@relcont
4657   \fi
4658   \ifx\eql@parseopt@token:%
4659     \let\eql@parseopt@next\eql@parseopt@relstart
4660   \fi
4661   \ifx\eql@parseopt@token|%
4662     \let\eql@parseopt@next\eql@parseopt@relord
4663   \fi
4664   \ifx\eql@parseopt@token?%
4665     \let\eql@parseopt@next\eql@columns@amp@parse@rel
4666   \fi
4667   \ifx\eql@parseopt@token&%
4668     \let\eql@parseopt@next\eql@parseopt@end
4669   \fi
4670 }
4671 \def\eql@columns@amp@parse@rel#1#2{%
4672   \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
4673   \eql@parseopt@end}

```

@columns@amp@process

```

4674 \def\eql@columns@amp@process{%
4675   \ifdefined\eql@punct@term\eql@punct@apply@top\fi
4676   \edef\eql@tmp{%
4677     \ifx\eql@class@rel@composed\@empty
4678       \ifodd\eql@column@&\noexpand\eql@punct@next@clear\fi&%
4679     \else

```

```

4680     \ifodd\eql@column@\else&\fi%
4681     \unexpanded\expandafter{\eql@class@rel@composed}%
4682     \fi
4683 }%
4684 \eql@tmp
4685 }

```

14.3 Transpose

TODO: describe

TODO: adjust to \&?!

TODO: describe

```

4686 \let\eql@transpose@active\eql@false
4687 \def\eql@transpose@end{\eql@transpose@end}
4688 \def\eql@transpose@skip{&\eql@punct@next@clear}
4689 \def\eql@transpose@complete{%
4690   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

TODO: describe

```

4691 \def\eql@transpose{%
4692   \eql@totalcolumns@z@
4693   \eql@totalrows@z@
4694   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&%
4695   \eql@scan@reg@{}%
4696   \eql@row@z@
4697   \eql@transpose@output@row
4698 }

```

TODO: describe

```

4699 \def\eql@transpose@save@col#1{%
4700   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
4701     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

TODO: describe

```

4702 \def\eql@transpose@scan@col#1&{%
4703   \def\eql@tmpa{#1}%
4704   \ifx\eql@tmpa\eql@transpose@end\else
4705     \advance\eql@totalcolumns@ \@ne
4706     \eql@row@z@
4707     \let\eql@transpose@data@col\@empty
4708     \eql@transpose@scan@row#1\\eql@transpose@end\\%
4709     \ifnum\eql@row@>\eql@totalrows@
4710       \eql@totalrows@\eql@row@
4711       \fi
4712     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
4713     \expandafter\eql@transpose@scan@col
4714   \fi
4715 }

```

TODO: describe

```

4716 \def\eql@transpose@append@row#1{%
4717   \advance\eql@row@\@ne
4718   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}

```

TODO: describe

```
4719 \def\eql@transpose@scan@row#1\\{%
4720   \def\eql@tmpa{#1}%
4721   \ifx\eql@tmpa\eql@transpose@end\else
4722     \ifx\eql@transpose@active+
4723       \eql@transpose@scan@cell#1&\eql@transpose@end&%
4724     \else
4725       \eql@transpose@append@row{#1}%
4726     \fi
4727     \expandafter\eql@transpose@scan@row
4728   \fi
4729 }
```

TODO: describe

```
4730 \def\eql@transpose@scan@cell#1&#2{%
4731   \def\eql@tmpa{#2}%
4732   \ifx\eql@tmpa\eql@transpose@end
4733     \eql@transpose@append@row{#1}%
4734   \else
4735     \eql@transpose@append@row{#1&#2}%
4736     \expandafter\eql@transpose@scan@cell@next
4737   \fi
4738 }
```

TODO: describe

```
4739 \def\eql@transpose@scan@cell@next#1&{%
4740   \def\eql@tmpa{#1}%
4741   \ifx\eql@tmpa\eql@transpose@end\else
4742     \eql@transpose@append@row{&#1}%
4743     \expandafter\eql@transpose@scan@cell@next
4744   \fi
4745 }
```

TODO: describe

```
4746 \def\eql@transpose@output@row{%
4747   \ifnum\eql@row@<\eql@totalrows@
4748     \advance\eql@row@\@ne
4749     \eql@column@\z@
4750     \eql@transpose@output@col
4751     \ifnum\eql@row@<\eql@totalrows@
4752       \eql@scan@addto\\%
4753     \fi
4754     \expandafter\eql@transpose@output@row
4755   \fi
4756 }
```

TODO: describe

```
4757 \def\eql@transpose@output@col{%
4758   \ifnum\eql@column@<\eql@totalcolumns@
4759     \advance\eql@column@\@ne
4760     \csname eql@transpose@data@col@\the\eql@column@\endcsname
4761     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4762     \ifnum\eql@column@<\eql@totalcolumns@
4763       \eql@scan@addto{\eql@transpose@complete}%
4764     \fi
4765     \expandafter\eql@transpose@output@col
```

```

4766 \fi
4767 }

```

14.4 Measure

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

```

s@measure@line@begin

```

```

4768 \def\eql@columns@measure@line@begin{%
4769   \eql@verbose@info\eql@verbose@msg@startline@number
4770   \global\eql@column@\z@
4771   \global\eql@line@height@\z@
4772   \global\eql@line@depth@\z@
4773   \eql@numbering@measure@line@begin
4774   \eql@hook@linein
4775 }

```

```

4776 \def\eql@columns@measure@cell{%
4777   \eql@cellwidth@\wd\eql@cellbox@
4778   \ifdefined\eql@frame@cmd
4779     \eql@frame@measure
4780     \advance\eql@cellwidth@\eql@frame@margin@
4781   \fi
4782   \ifdim\ht\eql@cellbox@>\eql@line@height@
4783     \global\eql@line@height@\ht\eql@cellbox@
4784   \fi
4785   \ifdim\dp\eql@cellbox@>\eql@line@depth@
4786     \global\eql@line@depth@\dp\eql@cellbox@
4787   \fi
4788   \ifnum\eql@column@=\@ne
4789     \eql@dimensions@startrow
4790   \fi
4791   \ifodd\eql@column@
4792     \eql@shape@pos@\tw@
4793   \else
4794     \eql@shape@pos@\z@
4795   \fi
4796   \eql@shape@amount@\z@
4797   \eql@dimensions@savecell
4798   \ifodd\eql@column@\else
4799     \eql@dimensions@savesep
4800   \fi
4801   \kern\eql@cellwidth@
4802 }

```

```

mns@measure@line@end

```

```

4803 \def\eql@columns@measure@line@end{%
4804   \eql@punct@apply@line
4805   \eql@hook@lineout
4806   &\omit
4807   \ifnum\eql@column@>\eql@totalcolumns@
4808     \global\eql@totalcolumns@\eql@column@
4809   \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

```

4810 \ifdefined\eql@frame@cmd
4811 \advance\eql@column@{\@ne
4812 \wd\eql@cellbox@z@
4813 \eql@columns@measure@cell
4814 \fi
4815 \eql@measure@tag
4816 \eql@measure@endrow
4817 }

```

\eql@columns@measure

```

4818 \def\eql@columns@measure{%
4819 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@measure
4820 \eql@totalcolumns@z@
4821 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end
4822 \ifdefined\eql@amp@mode
4823 \let\&\eql@columns@amp
4824 \fi

4825 \setbox\z@\vbox{\measuring@true\halign{%
4826 &%
4827 \global\advance\eql@column@{\@ne
4828 \global\let\eql@cell@container\@empty
4829 \global\setbox\eql@cellbox@\hbox{%
4830 \eql@strut@cell
4831 \@lign
4832 $\m@th\eql@mathstyle
4833 \eql@hook@colin
4834 ##%
4835 \eql@punct@apply@next
4836 \eql@class@innerleft
4837 \eql@hook@innerleft
4838 $%
4839 }%
4840 \eql@cell@container
4841 \hfil
4842 \eql@columns@measure@cell
4843 \global\let\eql@frame@prevcmd\eql@frame@cmd
4844 &%
4845 \eql@prevwidth@\wd\eql@cellbox@
4846 \let\eql@frame@cmd\eql@frame@prevcmd
4847 \global\advance\eql@column@{\@ne
4848 \global\let\eql@cell@container\@empty
4849 \setbox\eql@cellbox@\hbox{%
4850 \eql@strut@cell
4851 \@lign
4852 $\m@th\eql@mathstyle
4853 \eql@hook@innerright
4854 \eql@class@innerright@sel
4855 ##%
4856 \eql@punct@apply@col
4857 \eql@hook@colout
4858 $%
4859 }%
4860 \eql@cell@container
4861 \eql@columns@measure@cell
4862 \hfil

```

```

4863 \crrr

4864 \noalign{%
4865 \eql@hook@blockbefore
4866 }%
4867 \eql@hook@blockin
4868 \eql@scan@body

4869 \ifvmode\else
4870 \eql@punct@apply@block
4871 \eql@hook@blockout
4872 \eql@multi@endline
4873 \cr
4874 \fi
4875 \noalign{%
4876 \eql@hook@blockafter
4877 }%

```

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

```

4878 \omit
4879 \eql@column@\@ne
4880 \eql@columns@completerow
4881 \cr
4882 }}%

4883 \eql@measure@close

4884 \setbox\z@\vbox{%
4885 \unvbox\z@
4886 \unpenalty
4887 \global\setbox\@ne\lastbox
4888 }%
4889 \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4890 \let\eql@colwidth@tab\@empty
4891 \loop
4892 \setbox\@ne\hbox{%
4893 \unhbox\@ne
4894 \unskip
4895 \global\setbox\thr@\@ne\lastbox
4896 }%
4897 \ifhbox\thr@@
4898 \eql@colwidth@save{\wd\thr@@}%
4899 \repeat

4900 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@measure
4901 }

```

14.5 Columns Placement

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

```

4902 \def\eql@columns@adjust{%
4903 \ifodd\eql@totalcolumns@
4904 \advance\eql@totalcolumns@\@ne
4905 \fi

```

```

4906 \eq\tagpos@adjust@eval
4907 \eq\adjust@calc@columns
4908 }

```

14.6 Print

TODO: describe

```

mns@print@line@begin

```

```

4909 \def\eq\columns@print@line@begin{%
4910 \eq\verbose@info\eq\verbose@msg@startline@number
4911 \global\eq\column@\z@
4912 \global\eq\line@pos@\eq\marginleft@
4913 \global\eq\line@width@\z@
4914 \global\eq\line@avail@\eq\totalwidth@
4915 \global\eq\line@height@\z@
4916 \global\eq\line@depth@\z@
4917 \eq\numbering@print@line@begin
4918 \eq\hook@linein
4919 }

```

```

l@columns@print@cell

```

```

4920 \def\eq\columns@print@cell{%
4921 \eq\cellwidth@\wd\eq\cellbox@
4922 \ifodd\eq\column@
4923 \ifdefined\eq\frame@cmd
4924 \eq\frame@measure
4925 \advance\eq\cellwidth@\eq\frame@margin@
4926 \fi
4927 \dimen@\z@
4928 \else
4929 \advance\eq\cellwidth@-\eq\prevwidth@

```

draw a frame

```

4930 \ifdefined\eq\frame@cmd
4931 \eq\frame@measure
4932 \advance\eq\cellwidth@\eq\frame@margin@
4933 \advance\eq\prevwidth@\eq\frame@margin@
4934 \eq\frame@print
4935 \fi

```

update height and depth

```

4936 \ifdim\ht\eq\cellbox@>\eq\line@height@
4937 \global\eq\line@height@\ht\eq\cellbox@
4938 \fi
4939 \ifdim\dp\eq\cellbox@>\eq\line@depth@
4940 \global\eq\line@depth@\dp\eq\cellbox@
4941 \fi

```

print box

```

4942 \kern-\eq\prevwidth@
4943 \unhbox\eq\cellbox@
4944 \dimen@-\eq\cellwidth@
4945 \fi

```


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

```
4946 \advance\dimen@ \eq@colwidth@get\eq@column@ \relax
4947 \kern\dimen@
```

update available and used space

```
4948 \dimen@ \eq@colwidth@get\eq@column@ \relax
4949 \ifdim\eq@cellwidth@>\z@
4950 \ifdim\eq@line@width@=\z@
4951 \eq@line@avail@ \eq@line@pos@
4952 \ifodd\eq@column@
4953 \advance\eq@line@avail@ \dimen@
4954 \advance\eq@line@avail@ -\eq@cellwidth@
4955 \fi
4956 \global\eq@line@avail@ \eq@line@avail@
4957 \fi
4958 \eq@line@width@ \eq@line@pos@
4959 \ifodd\eq@column@
4960 \advance\eq@line@width@ \dimen@
4961 \else
4962 \advance\eq@line@width@ \eq@cellwidth@
4963 \fi
4964 \global\eq@line@width@ \eq@line@width@
4965 \fi
4966 \advance\eq@line@pos@ \dimen@
4967 \ifodd\eq@column@ \else
4968 \advance\eq@line@pos@ \eq@colsep@
4969 \fi
4970 \global\eq@line@pos@ \eq@line@pos@
4971 }

4972 \def\eq@columns@print@trailright{%
4973 &\omit
4974 \eq@prevwidth@ \wd\eq@cellbox@
4975 \let\eq@frame@cmd\eq@frame@prevcmd
4976 \global\advance\eq@column@ \@ne
4977 \eq@columns@print@cell
4978 }
```

lums@print@line@end **TODO:** add an even column with empty stuff if box processing deferred

```
4979 \def\eq@columns@print@line@end{%
4980 \eq@punct@apply@line
4981 \eq@hook@lineout
4982 \ifodd\eq@column@
4983 \expandafter\eq@columns@print@trailright
4984 \fi
4985 \eq@columns@completerow
4986 \eq@columns@print@tag
4987 }
```

ql@columns@print@tag

```
4988 \def\eq@columns@print@tag{%
4989 \kern-\dimexpr\eq@totalwidth@+\eq@colsep@ \relax
```

determine first line available space

```

4990 \eqldisplay@firstavail@set\eql@line@avail@
4991 \eql@columns@overfull
4992 \eql@numbering@print@line@eval
4993 \if@eqnsw
4994 \eql@tagbox@make\eql@composetag@print
4995 \fi
4996 \eql@tagpos@print@line@eval
4997 \eql@tagbox@print@cell
4998 \eql@tagpos@print@line@end
4999 }

```

\eql@columns@print

```

5000 \def\eql@columns@print{%
5001 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
5002 \eql@shape@align@disable
5003 \eqldisplay@halign@init\eql@columns@print@line@begin
5004 \eql@multi@cr@let\eql@columns@print@line@end
5005 \ifdefined\eql@amp@mode
5006 \let\&\eql@columns@amp
5007 \fi
5008 \tabskip\eql@marginleft@

5009 \halign{%
5010 &%
5011 \global\advance\eql@column@\@ne
5012 \global\let\eql@cell@container\@empty
5013 \global\setbox\eql@cellbox@\hbox{%
5014 \eql@strut@cell
5015 \@lign
5016 $\m@th\eql@mathstyle
5017 \eql@hook@colin
5018 ##%
5019 \eql@punct@apply@next
5020 \eql@class@innerleft
5021 \eql@hook@innerleft
5022 \eql@tagging@mathsave
5023 $%
5024 \eql@tagging@mathaddlast
5025 }%
5026 \eql@cell@container
5027 \hfil
5028 \eql@columns@print@cell
5029 \global\let\eql@frame@prevcmd\eql@frame@cmd
5030 \tabskip\z@skip
5031 &%
5032 \eql@prevwidth@\wd\eql@cellbox@
5033 \let\eql@frame@cmd\eql@frame@prevcmd
5034 \global\advance\eql@column@\@ne
5035 \global\let\eql@cell@container\@empty
5036 \setbox\eql@cellbox@\hbox{%
5037 \unhbox\eql@cellbox@
5038 \eql@strut@cell
5039 \@lign
5040 $\m@th\eql@mathstyle
5041 \eql@hook@innerright
5042 \eql@class@innerright@sel
5043 ##%
5044 \eql@punct@apply@col

```

```

5045         \eql@hook@colout
5046         \eql@tagging@mathsave
5047         $%
5048         \eql@tagging@mathaddlast
5049     }%
5050     \eql@cell@container
5051     \eql@columns@print@cell
5052     \hfil
5053     \tabskip\eql@colsep@\relax
5054 \crrr

5055 \noalign{%
5056     \eql@display@halign@start
5057     \eql@numbering@print@block@begin
5058     \eql@hook@blockbefore
5059 }%
5060 \eql@hook@blockin
5061 \eql@scan@body
5062 \ifvmode\else
5063     \relax
5064     \eql@punct@apply@block
5065     \eql@hook@blockout
5066     \eql@multi@endline
5067     \cr
5068 \fi
5069 \noalign{%
5070     \eql@hook@blockafter
5071     \eql@display@halign@end
5072     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
5073 }%
5074 \eql@tagging@tablesalign
5075 }%
5076 }

```

15 Interface

15.1 Options Processing

TODO: describe

```

5077 \def\eql@equations@test@setopt{\let\eql@equations@test\eql@equations@testopt}
5078 \def\eql@equations@test@setall{\let\eql@equations@test\eql@equations@testall}

```

`\eql@equations@testopt` **TODO:** describe

```

5079 \def\eql@equations@testopt#1{%
5080     \eql@testopt@tight{\eql@equations@testopt@set{#1}}{}}
5081 \def\eql@equations@testopt@set#1[#2]{\eqnaddopt{#2}#1}

```

`\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

```

5082 \def\eql@equations@testall{\eql@parseopt@env\eql@equations@testall@parse}
5083 \def\eql@equations@testall@parse{%
5084     \ifx\eql@parseopt@token*%

```

```

5085 \let\eql@parseopt@next\eql@parseopt@nonumber
5086 \fi
5087 \ifx\eql@parseopt@token!%
5088 \let\eql@parseopt@next\eql@parseopt@donumber
5089 \fi
5090 \ifx\eql@parseopt@token/%
5091 \let\eql@parseopt@next\eql@parseopt@transpose
5092 \fi
5093 \ifx\eql@parseopt@token[%]
5094 \let\eql@parseopt@next\eql@parseopt@opt
5095 \fi
5096 \ifx\eql@parseopt@token\eql@atxi
5097 \let\eql@parseopt@next\eql@parseopt@label
5098 \fi
5099 \ifx\eql@parseopt@token\eql@atxii
5100 \let\eql@parseopt@next\eql@parseopt@label
5101 \fi
5102 \ifx\eql@parseopt@token.%
5103 \let\eql@parseopt@next\eql@parseopt@punctpass
5104 \fi
5105 \ifx\eql@parseopt@token,%
5106 \let\eql@parseopt@next\eql@parseopt@punctpass
5107 \fi
5108 \ifx\eql@parseopt@token~%
5109 \let\eql@parseopt@next\eql@parseopt@punctpass
5110 \fi
5111 \ifx\eql@parseopt@token'%
5112 \let\eql@parseopt@next\eql@parseopt@punctopt
5113 \fi
5114 \ifx\eql@parseopt@token-%
5115 \let\eql@parseopt@next\eql@parseopt@single
5116 \fi
5117 \ifx\eql@parseopt@token=%
5118 \let\eql@parseopt@next\eql@parseopt@lines
5119 \fi
5120 \ifx\eql@parseopt@token|%
5121 \let\eql@parseopt@next\eql@parseopt@columns
5122 \fi
5123 \ifx\eql@parseopt@token<%
5124 \let\eql@parseopt@next\eql@parseopt@ampeq
5125 \fi
5126 \ifx\eql@parseopt@token>%
5127 \let\eql@parseopt@next\eql@parseopt@eqamp
5128 \fi
5129 \ifx\eql@parseopt@token\label
5130 \let\eql@parseopt@next\eql@parseopt@end
5131 \fi
5132 \ifx\eql@parseopt@token\begin
5133 \let\eql@parseopt@next\eql@parseopt@end
5134 \fi
5135 }

```

equations@end@testall **TODO:** describe

```

5136 \def\eql@equations@end@testall{%
5137 \eql@parseopt@env\eql@equations@end@testall@parse}
5138 \def\eql@equations@end@testall@parse{%
5139 \ifx\eql@parseopt@token.%
5140 \let\eql@parseopt@next\eql@parseopt@punctpass

```

```

5141 \fi
5142 \ifx\eql@parseopt@token,%
5143   \let\eql@parseopt@next\eql@parseopt@punctpass
5144 \fi
5145 \ifx\eql@parseopt@token~%
5146   \let\eql@parseopt@next\eql@parseopt@punctpass
5147 \fi
5148 \ifx\eql@parseopt@token'%
5149   \let\eql@parseopt@next\eql@parseopt@punctblock
5150 \fi
5151 }

```

`\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:

```

5152 \def\eql@equations@processopt{%
5153   \let\eql@tags@container@block\eql@tags@container@clear
5154   \let\eql@tags@frame@cmd\@firstofone
5155   \let\eql@skip@force@above\@undefined
5156   \let\eql@skip@force@below\@undefined
5157   \let\eql@skip@force@leave\@undefined
5158   \let\eql@display@linewidth\@undefined
5159   \let\eql@display@marginleft\@undefined
5160   \let\eql@display@marginright\@undefined
5161   \eql@abovespace@\z@skip
5162   \eql@belowspace@\z@skip
5163   \eql@displaybreak@prepen@\@MM
5164   \eql@displaybreak@postpen@\@MM
5165   \eql@nextopt@process{equations}%
5166   \let\eql@punct@next\@undefined
5167   \eql@indent@\glueexpr\eql@indent@val\relax
5168   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
5169 }

```

15.2 Single-Line Main

In the following, we define the main routine for the single-line equation mode.

`\eql@single@cr@error` Cannot use line breaks, produce an error message:

```

5170 \def\eql@single@cr@error{%
5171   \eql@error{Cannot use '\string\\' within display equation.
5172     Please switch to equations environment}%
5173 }

```

`\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:

```

5174 \def\eql@single@start{%
5175   \eql@display@enter
5176   \eql@tagging@start
5177   \eql@dollar@dollar@begin
5178   \eql@display@adjust

```

```

5179 \eql@numbering@init
5180 \eql@stack@save@equations
5181 \eql@numbering@single@init
5182 \ifdefined\eql@single@cr@mode
5183   \let\\\eql@single@cr@mode
5184 \fi
5185 \ifdefined\eql@amp@mode
5186   \let\&\eql@break@amp
5187 \fi
5188 \ifdefined\eql@single@native
5189   \let\eql@single@start@sel\eql@single@start@native
5190   \let\eql@single@end@sel\eql@single@end@native
5191 \else
5192   \let\eql@single@start@sel\eql@single@start@print
5193   \let\eql@single@end@sel\eql@single@end@print
5194 \fi
5195 \eql@single@start@sel
5196 }

```

`\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:

```

5197 \def\eql@single@end{%
5198   \eql@punct@apply@block
5199   \eql@hook@eqout
5200   \eql@single@end@sel
5201   \global\eql@punct@top@reset
5202   \eql@stack@restore
5203   \eql@dollar@end
5204   \eql@tagging@end
5205   \eql@display@leave
5206 }

```

`\eql@single@main` Combined opening, body and closing for pre-scanned body:

```

5207 \def\eql@single@main{%
5208   \eql@single@start
5209   \eql@scan@body
5210   \eql@single@end
5211 }

```

`equations@single@set` Configure equations macros to single-line mode:

```

5212 \def\eql@equations@single@set{%
5213   \ifdefined\eql@single@doscan
5214     \let\eql@equations@main\eql@single@main
5215   \else
5216     \let\eql@equations@main\@undefined
5217   \fi
5218 }

```

15.3 Multi-Line Main

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

```

5219 \let\eql@multi@linesmode\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

```

5220 \def\eql@multi@main{%
5221   \eql@display@enter
5222   \eql@tagging@start
5223   \eql@dollar@begin
5224   \eql@display@adjust
5225   \eql@numbering@init
5226   \eql@stack@save@equations
5227   \ifdefined\eql@transpose@active
5228     \ifdefined\eql@multi@linesmode\else
5229       \eql@transpose
5230     \fi
5231   \fi
5232   \ifdefined\eql@numbering@subeq@use
5233     \eql@numbering@subeq@init
5234   \fi
5235   \eql@display@init
5236   \let\intertext\eql@intertext
5237   \let\endintertext\endeql@intertext
5238   \eql@shape@align@enable

```

Now measure the given multi-line equations body:

```

5239   \ifdefined\eql@multi@linesmode
5240     \eql@lines@measure
5241   \else
5242     \ifdefined\eql@ampproof@active
5243       \eql@ampproof
5244     \fi
5245     \eql@columns@measure
5246   \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`:

```

5247   \ifdefined\eql@numbering@subeq@use
5248     \eql@numbering@subeq@test
5249   \fi
5250   \ifdefined\eql@multi@linesmode\else
5251     \ifdefined\eql@multi@linesfallback
5252       \ifnum\eql@totalcolumns@=\@ne
5253         \let\eql@multi@linesmode\eql@true
5254         \ifx\eql@multi@linesfallback\z@\else
5255           \eql@lines@measure
5256         \fi
5257       \fi
5258     \fi
5259   \fi

```

Adjust the multi-line equations body:

```

5260   \ifdefined\eql@multi@linesmode
5261     \eql@lines@adjust
5262   \else

```

```

5263 \eql@columns@adjust
5264 \fi

```

Now print the multi-line equations body:

```

5265 \eql@display@print
5266 \eql@numbering@print@init
5267 \ifdefined\eql@multi@linesmode
5268 \eql@lines@print
5269 \else
5270 \eql@columns@print
5271 \fi
5272 \eql@display@close

```

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

```

5273 \ifdefined\eql@numbering@subeq@use
5274 \eql@numbering@subeq@close
5275 \fi
5276 \global\eql@punct@top@reset
5277 \eql@stack@restore
5278 \eql@dollar@dollar@end
5279 \eql@tagging@end
5280 \eql@display@leave
5281 }

```

`equations@columns@set` Configure equations macros to one of the two multi-line modes:
`@equations@lines@set`

```

5282 \def\eql@equations@columns@set{%
5283 \let\eql@equations@main\eql@multi@main
5284 \let\eql@multi@linesmode\eql@false
5285 }
5286 \def\eql@equations@lines@set{%
5287 \let\eql@equations@main\eql@multi@main
5288 \let\eql@multi@linesmode\eql@true
5289 }

```

15.4 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`:

```

5290 \newenvironment{equations}{%
5291 \ifmmode
5292 \expandafter\eql@equations@env@cancel
5293 \else
5294 \eql@verbose@info\eql@verbose@msg@enterenv
5295 \expandafter\eql@equations@env@open
5296 \fi
5297 }{%
5298 \ifdefined\eql@equations@main\else
5299 \expandafter\eql@single@end

```



```

5300 \fi
5301 \ignorespacesafterend
5302 \eql@verbose@info\eql@verbose@msg@leaveenv
5303 }
5304 \eql@markline@amsthm@register{equations}
5305 \eql@tagging@register@luamml{equations}

```

equations@env@cancel

```

5306 \def\eql@equations@env@cancel{%
5307 \eql@error@mathmode{\string\begin{\@currenvir}}}%
5308 \let\eql@scan@call\eql@scan@env@cancel
5309 \eql@scan@env
5310 }

```

l@equations@env@open

```

5311 \def\eql@equations@env@open{%
5312 \ifdefined\eql@equations@env@modifier
5313 \eql@equations@test@setall
5314 \else
5315 \eql@equations@nomodifier
5316 \fi
5317 \eql@ampprotect\eql@equations@test\eql@equations@env@start
5318 }

```

@equations@env@start The macro `\eql@equations@env@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`:

```

5319 \def\eql@equations@env@start{%
5320 \eql@equations@processopt
5321 \ifdefined\eql@equations@main
5322 \eql@equations@call@set
5323 \expandafter\eql@scan@env
5324 \else
5325 \expandafter\eql@single@start
5326 \fi
5327 }

```

\eql@equations@call

```

5328 \def\eql@equations@call{\eql@equations@main\eql@scan@end}
5329 \def\eql@equations@call@test{%
5330 \eql@ampprotect\eql@equations@end@testall\eql@equations@call}
5331 \def\eql@equations@call@set{%
5332 \ifdefined\eql@equations@end@modifier
5333 \let\eql@scan@call\eql@equations@call@test
5334 \else
5335 \let\eql@scan@call\eql@equations@call
5336 \fi
5337 }

```

Square Brackets.

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

```

5338 \newenvironment{equations@sqr}{\}{}
5339 \eq@markline@amsthm@register{equations@sqr}
5340 \eq@tagging@register@luamml{equations@sqr}

```

`l@equations@sqr@open` Definition for ‘`\[`’. Add the default arguments `\eq@equations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eq@equations@sqr@start`:

```

5341 \def\eq@equations@sqr@open{%
5342   \expandafter\eqnaddopt\expandafter{\eq@equations@sqr@opt}%
5343   \begin{equations@sqr}%
5344   \eq@verbose@info\eq@verbose@msg@enterenv
5345   \let\]\eq@equations@sqr@close
5346   \ifdefined\eq@equations@sqr@modifier
5347     \eq@equations@test@setall
5348   \else
5349     \eq@equations@nomodifier
5350   \fi
5351   \eq@ampprotect\eq@equations@test\eq@equations@sqr@start
5352 }

```

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

```

5353 \def\eq@equations@sqr@start{%
5354   \eq@equations@processopt
5355   \ifdefined\eq@equations@main
5356     \eq@equations@call@set
5357     \expandafter\eq@scan@sqr
5358   \else
5359     \expandafter\eq@single@start
5360   \fi
5361 }

```

`@equations@sqr@close` Definition for ‘`\]`’. Parse modifiers following ‘`\]`’ and hand on to `\eq@equations@sqr@end`:

```

5362 \protected\def\eq@equations@sqr@close{%
5363   \ifdefined\eq@equations@main
5364     \let\eq@equations@end@modifier\eq@false
5365   \fi
5366   \ifdefined\eq@equations@end@modifier
5367     \expandafter\eq@ampprotect\expandafter\eq@equations@end@testall
5368   \fi
5369   \eq@equations@sqr@end
5370 }

```

`ql@equations@sqr@end` **TODO:** complete End `\[...]` block:

```

5371 \def\eq@equations@sqr@end{%
5372   \ifdefined\eq@equations@main\else
5373     \expandafter\eq@single@end
5374   \fi
5375   \eq@verbose@info\eq@verbose@msg@leaveenv
5376   \end{equations@sqr}%
5377   \ignorespaces
5378 }

```

Angle Brackets.

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

```

5379 \newenvironment{equations@ang}{\{}}{}
5380 \eq@markline@amsthm@register{equations@ang}
5381 \eq@tagging@register@luamml{equations@ang}

\equations@ang@open

5382 \def\eq@equations@ang@open{%
5383   \expandafter\eqnaddopt\expandafter{\eq@equations@ang@opt}%
5384   \begin{equations@ang}%
5385   \eq@verbose@info\eq@verbose@msg@enterenv
5386   \let\>\eq@equations@ang@close
5387   \ifdefined\eq@equations@ang@modifier
5388     \eq@equations@test@setall
5389   \else
5390     \eq@equations@nomodifier
5391   \fi
5392   \eq@ampprotect\eq@equations@test\eq@equations@ang@start
5393 }
```

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

```

5394 \def\eq@equations@ang@start{%
5395   \eq@equations@processopt
5396   \ifdefined\eq@equations@main
5397     \eq@equations@call@set
5398     \expandafter\eq@scan@ang
5399   \else
5400     \expandafter\eq@single@start
5401   \fi
5402 }
```

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

```

5403 \def\eq@equations@ang@close{%
5404   \ifdefined\eq@equations@main
5405     \let\eq@equations@end@modifier\eq@false
5406   \fi
5407   \ifdefined\eq@equations@end@modifier
5408     \expandafter\eq@ampprotect\expandafter\eq@equations@end@testall
5409   \fi
5410   \eq@equations@ang@end
5411 }
```

`ql@equations@ang@end` **TODO:** describe

```

5412 \def\eq@equations@ang@end{%
5413   \ifdefined\eq@equations@main\else
5414     \expandafter\eq@single@end
5415   \fi
5416   \eq@verbose@info\eq@verbose@msg@leaveenv
5417   \end{equations@ang}%
5418   \ignorespaces
5419 }
```

16 Options

16.1 Selection Tools

`\eqldecide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```
5420 \def\eqldecide@abovebelow#1#2#3#4#5{%
5421   \eqldecide@select{#1}{#2}{#3}{%
5422     {abovebelow,both,tb}{#4#5},%
5423     {above,top,t}{#4},%
5424     {below,bottom,b}{#5}}}
```

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

```
5425 \def\eqldecide@situation#1#2#3#4{%
5426   \eqldecide@select{#1}{#2}{#3}{%
5427     {{long}{\def#4{0}}},%
5428     {{short}{\def#4{1}}},%
5429     {{cont}{\def#4{2}}},%
5430     {{par}{\def#4{3}}},%
5431     {{top}{\def#4{4}}},%
5432     {{noskip}{\def#4{5}}},%
5433     {{medskip}{\def#4{6}}}}}
```

`\eqldecide@delim` **TODO:** describe

```
5434 \def\eqldecide@delim#1#2#3{%
5435   \eqldecide@select{#1}{#2}{#3}{%
5436     {{,.,\eqldecide@false}{\eqld@box@wrap{}{}}},%
5437     {{\eqldecide@true,r,round}{\eqld@box@delim()}},%
5438     {{s,sqr,square}{\eqld@box@delim[]}},%
5439     {{c,curly,braces}{\eqld@box@delim\lbrace\rbrace}},%
5440     {{a,ang,angle}{\eqld@box@delim\langle\rangle}},%
5441     {{v,vert}{\eqld@box@delim\vert\vert}},%
5442     {{d,dvert}{\eqld@box@delim\Vert\Vert}},%
5443     {\relax{\eqld@box@delim#3}}}%}
```

TODO: describe

```
5444 \def\eqld@keyall{equations,box,setup}
```

16.2 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:

```
5445 \eqld@define@key{box}{lines,ln,gathered,gather,ga}[]{%
5446   \eqld@box@lines@set}
5447 \eqld@define@key{box}{columns,col,aligned,align,al}[]{%
5448   \eqld@box@columns@set}
5449 \eqld@define@key{box}{cases}[]{%
5450   \eqld@box@cases@set\eqld@box@ldelim\lbrace%
5451   \def\eqld@box@colsep{\eqld@box@condsep}}
5452 \eqld@define@key{box}{matrix}[r]{%
```

```

5453 \eql@box@lines@set\eql@shape@set{center}%
5454 \let\eql@spread@reset\eql@true\def\eql@spread@val{\z@}%
5455 \def\eql@box@colsep{\eql@box@matrixsep}%
5456 \let\eql@mathstyle\@empty
5457 \eql@punct@clear
5458 \eql@box@cr@test@setopt
5459 \eql@box@amp@test@setopt
5460 \let\eql@box@end@modifier\eql@false
5461 \eql@decide@delim{#3}{#2}{#1}}
5462 \eql@define@key{box}{top,t}[]{\let\eql@box@box\vtop}
5463 \eql@define@key{box}{center,c}[]{\let\eql@box@box\vcenter}
5464 \eql@define@key{box}{bottom,b}[]{\let\eql@box@box\vbbox}
5465 \eql@define@key{box}{intro}{%
5466 \def\eql@box@cases@condintro{#1}}
5467 \eql@define@key{box}{introtext}{%
5468 \def\eql@box@cases@condintro{%
5469 \ifmmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}
5470 \eql@define@key{box}{textcond}[true]{%
5471 \eql@decide@select{#3}{#2}{#1}{%
5472 {\eql@decide@true,text}{\let\eql@box@cases@condtext\eql@true}},%
5473 {\eql@decide@false,math}{\let\eql@box@cases@condtext\eql@false}}}%
5474 \eql@define@key{setup}{scanbox}[true]{%
5475 \eql@decide@bool{#3}{#2}{#1}\eql@box@doscanscan}
5476 \eql@define@key{box}{scan}[true]{%
5477 \eql@decide@bool{#3}{#2}{#1}\eql@box@doscanscan}
5478 \eql@define@key{setup}{boxangopt}[]{%
5479 \def\eql@box@ang@opt{columns,#1}}

```

Modes for Equations Environment.

```
5480 \let\eql@box@doscanscan\eqn@false
```

Declare modes and switches for the equations environment:

```

5481 \eql@define@key{equations}{single,1,equation,eq}[]{\eql@equations@single@set}
5482 \eql@define@key{equations}{lines,ln,gathered,gather,ga}[]{%
5483 \eql@equations@lines@set}
5484 \eql@define@key{equations}{columns,col,aligned,align,al}[]{%
5485 \eql@equations@columns@set}
5486 \eql@define@key{equations,setup}{transpose}[true]{%
5487 \eql@decide@select{#3}{#2}{#1}{%
5488 {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
5489 {\noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
5490 {\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}%
5491 \eql@define@key{setup}{nativeequation}[true]{%
5492 \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
5493 \eql@define@key{equations}{native}[true]{%
5494 \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
5495 \ifdefined\eql@single@native\let\eql@layoutright\eql@false\fi}
5496 \eql@define@key{setup}{scanequation}[true]{%
5497 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscanscan}
5498 \eql@define@key{equations}{scan}[true]{%
5499 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscanscan}
5500 \eql@define@key{setup}{sqropt}[]{%
5501 \def\eql@equations@sqr@opt{equation,#1}}
5502 \eql@define@key{setup}{angopt}[]{%
5503 \def\eql@equations@ang@opt{columns,#1}}

```

TODO: describe

```

5504 \eqld@define@key{control}{restoreexterior}[]{\eqldisplay@restore}
5505 \eqld@define@key{control}{restoreexterior*}[]{\@arrayparboxrestore}

```

Modes for Modifier Scanning.

```

5506 \def\eqld@equations@nomodifier{%
5507   \eqld@equations@test@setopt
5508   \let\eqld@equations@end@modifier\eqld@false
5509   \eqld@multi@cr@test@setopt
5510   \eqld@columns@amp@test@setopt
5511   \eqld@break@cr@test@setopt
5512   \eqld@break@amp@test@setopt
5513 }
5514 \let\eqld@equations@env@modifier\eqld@false
5515 \let\eqld@equations@sqr@modifier\eqld@true
5516 \let\eqld@equations@ang@modifier\eqld@true
5517 \let\eqld@equations@end@modifier\eqld@true
5518 \eqld@equations@test@setall
5519 \eqld@multi@cr@test@setall
5520 \eqld@columns@amp@test@setall
5521 \eqld@break@cr@test@setall
5522 \eqld@break@amp@test@setall
5523 \eqld@break@test@setopt
5524 \let\eqld@multi@cr@relnext\eqld@true

5525 \def\eqld@box@nomodifier{%
5526   \eqld@box@test@setopt
5527   \let\eqld@box@end@modifier\eqld@false
5528   \eqld@box@cr@test@setopt
5529   \eqld@box@amp@test@setopt
5530 }
5531 \let\eqld@box@env@modifier\eqld@false
5532 \let\eqld@box@ang@modifier\eqld@true
5533 \let\eqld@box@end@modifier\eqld@true
5534 \eqld@box@test@setall
5535 \eqld@box@cr@test@setall
5536 \eqld@box@amp@test@setall

5537 \eqld@define@key{setup}{modifier}[true]{%
5538   \eqld@decide@if{#3}{#2}{#1}%
5539   {\let\eqld@equations@env@modifier\eqld@true
5540    \let\eqld@equations@sqr@modifier\eqld@true
5541    \let\eqld@equations@ang@modifier\eqld@true
5542    \let\eqld@box@env@modifier\eqld@true
5543    \let\eqld@box@ang@modifier\eqld@true
5544    \eqld@multi@cr@test@setall\eqld@break@cr@test@setall
5545    \eqld@columns@amp@test@setall\eqld@break@amp@test@setall
5546    \eqld@box@cr@test@setall\eqld@box@amp@test@setall
5547    \eqld@break@test@setall}%
5548   {\let\eqld@equations@env@modifier\eqld@false
5549    \let\eqld@equations@sqr@modifier\eqld@false
5550    \let\eqld@equations@ang@modifier\eqld@false
5551    \let\eqld@box@env@modifier\eqld@false
5552    \let\eqld@box@ang@modifier\eqld@false
5553    \eqld@multi@cr@test@setopt\eqld@break@cr@test@setopt
5554    \eqld@columns@amp@test@setopt\eqld@break@amp@test@setopt
5555    \eqld@box@cr@test@setopt\eqld@box@amp@test@setopt
5556    \eqld@break@test@setopt}}
5557 \eqld@define@key{setup}{modifierenv}[true]{%

```

```

5558 \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier
5559 \eq@decide@bool{#3}{#2}{#1}\eq@box@env@modifier}
5560 \eq@define@key{setup}{modifiersqrang}[true]{%
5561 \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier
5562 \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier
5563 \eq@decide@bool{#3}{#2}{#1}\eq@box@ang@modifier}
5564 \eq@define@key{setup}{modifierend}[true]{%
5565 \eq@decide@bool{#3}{#2}{#1}\eq@equations@end@modifier
5566 \eq@decide@bool{#3}{#2}{#1}\eq@box@end@modifier}
5567 \eq@define@key{setup}{modifiereqncr}[true]{\eq@decide@if{#3}{#2}{#1}%
5568 {\eq@multi@cr@test@setall\eq@break@cr@test@setall
5569 \eq@box@cr@test@setall}}%
5570 {\eq@multi@cr@test@setopt\eq@break@cr@test@setopt
5571 \eq@box@cr@test@setopt}}
5572 \eq@define@key{setup}{modifiereqnamp}[true]{\eq@decide@if{#3}{#2}{#1}%
5573 {\eq@columns@amp@test@setall\eq@break@amp@test@setall
5574 \eq@box@amp@test@setall}}%
5575 {\eq@columns@amp@test@setopt\eq@break@amp@test@setopt
5576 \eq@box@amp@test@setopt}}
5577 \eq@define@key{setup}{modifierbreak}[true]{\eq@decide@if{#3}{#2}{#1}%
5578 \eq@break@test@setall\eq@break@test@setopt}
5579 \eq@define@key{setup}{modifierwarning}[all]{%
5580 \eq@decide@select{#3}{#2}{#1}}%
5581 {\eq@decide@false{\let\eq@parseopt@warn@env\empty
5582 \let\eq@parseopt@warn@cr\empty}},%
5583 {\env,-}{\let\eq@parseopt@warn@env\eq@warn@parseopt
5584 \let\eq@parseopt@warn@cr\empty}},%
5585 {\all,\eq@decide@true}{\let\eq@parseopt@warn@env\eq@warn@parseopt
5586 \let\eq@parseopt@warn@cr\eq@warn@parseopt}},%
5587 {\verbose,+}{\let\eq@parseopt@warn@env\eq@warn@parseopt@verbose
5588 \let\eq@parseopt@warn@cr\eq@warn@parseopt@verbose}}}

5589 \eq@define@key{setup}{modifiereqn}[true]{%
5590 \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier
5591 \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier
5592 \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier}
5593 \eq@define@key{setup}{modifiereqnenv}[true]{%
5594 \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier}
5595 \eq@define@key{setup}{modifiereqnsqr}[true]{%
5596 \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier}
5597 \eq@define@key{setup}{modifiereqnang}[true]{%
5598 \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier}
5599 \eq@define@key{setup}{modifiereqnend}[true]{%
5600 \eq@decide@bool{#3}{#2}{#1}\eq@equations@end@modifier}
5601 \eq@define@key{setup}{modifiereqncr}[true]{\eq@decide@if{#3}{#2}{#1}%
5602 {\eq@multi@cr@test@setall\eq@break@cr@test@setall}}%
5603 {\eq@multi@cr@test@setopt\eq@break@cr@test@setopt}}
5604 \eq@define@key{setup}{modifiereqnamp}[true]{\eq@decide@if{#3}{#2}{#1}%
5605 {\eq@columns@amp@test@setall\eq@break@amp@test@setall}}%
5606 {\eq@columns@amp@test@setopt\eq@break@amp@test@setopt}}

5607 \eq@define@key{setup}{modifierbox}[true]{%
5608 \eq@decide@bool{#3}{#2}{#1}\eq@box@env@modifier
5609 \eq@decide@bool{#3}{#2}{#1}\eq@box@ang@modifier}
5610 \eq@define@key{setup}{modifierboxenv}[true]{%
5611 \eq@decide@bool{#3}{#2}{#1}\eq@box@env@modifier}
5612 \eq@define@key{setup}{modifierboxang}[true]{%
5613 \eq@decide@bool{#3}{#2}{#1}\eq@box@ang@modifier}
5614 \eq@define@key{setup}{modifierboxend}[true]{%

```

```

5615 \eqld@decide@bool{#3}{#2}{#1}\eqld@box@end@modifier}
5616 \eqld@define@key{setup}{modifierboxcr}[true]{%
5617 \eqld@decide@if{#3}{#2}{#1}\eqld@box@cr@test@setall\eqld@box@cr@test@setopt}
5618 \eqld@define@key{setup}{modifierboxamp}[true]{%
5619 \eqld@decide@if{#3}{#2}{#1}\eqld@box@amp@test@setall\eqld@box@amp@test@setopt}

5620 \eqld@define@key{equations}{modifierend}[true]{%
5621 \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@end@modifier}
5622 \eqld@define@key{equations}{modifiercr}[true]{\eqld@decide@if{#3}{#2}{#1}%
5623 {\eqld@multi@cr@test@setall\eqld@break@cr@test@setall}%
5624 {\eqld@multi@cr@test@setopt\eqld@break@cr@test@setopt}}
5625 \eqld@define@key{equations}{modifieramp}[true]{\eqld@decide@if{#3}{#2}{#1}%
5626 {\eqld@columns@amp@test@setall\eqld@break@amp@test@setall}%
5627 {\eqld@columns@amp@test@setopt\eqld@break@amp@test@setopt}}
5628 \eqld@define@key{setup,equations}{crrelnext}[true]{%
5629 \eqld@decide@bool{#3}{#2}{#1}\eqld@multi@cr@relnext}

5630 \eqld@define@key{box}{modifier}[true]{\eqld@decide@if{#3}{#2}{#1}%
5631 {\eqld@box@cr@test@setall\eqld@box@amp@test@setall
5632 \let\eqld@box@end@modifier\eqld@true}%
5633 {\eqld@box@cr@test@setopt\eqld@box@amp@test@setopt
5634 \let\eqld@box@end@modifier\eqld@false}}
5635 \eqld@define@key{box}{modifierend}[true]{%
5636 \eqld@decide@bool{#3}{#2}{#1}\eqld@box@end@modifier}
5637 \eqld@define@key{box}{modifiercr}[true]{%
5638 \eqld@decide@if{#3}{#2}{#1}\eqld@box@cr@test@setall\eqld@box@cr@test@setopt}
5639 \eqld@define@key{box}{modifieramp}[true]{%
5640 \eqld@decide@if{#3}{#2}{#1}\eqld@box@amp@test@setall\eqld@box@amp@test@setopt}

```

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

```

5641 \eqld@define@key\eqld@keyall{spread}{%
5642 \let\eqld@spread@reset\eqld@false\def\eqld@spread@val{#1}}
5643 \eqld@define@key\eqld@keyall{spread*}[Opt]{%
5644 \let\eqld@spread@reset\eqld@true\def\eqld@spread@val{#1}}
5645 \eqld@define@key\eqld@keyall{strut}[true]{\eqld@decide@select{#3}{#2}{#1}{%
5646 {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%
5647 {{cell}{\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\relax}},%
5648 {{tag}{\let\eqld@strut@cell\relax\let\eqld@strut@tag\eqld@strut}},%
5649 {\eqld@decide@true
5650 {\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\eqld@strut}}}}
5651 \eqld@define@key{setup}{strutdepth}{\def\eqld@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

5652 \eqld@define@key\eqld@keyall{displayheight}[strut]{%
5653 \eqld@decide@select{#3}{#2}{#1}{%
5654 {\eqld@decide@false{\let\eqld@display@height\@undefined}},%
5655 {{strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
5656 {\relax{\def\eqld@display@height{#1}}}}}
5657 \eqld@define@key\eqld@keyall{displaydepth}[strut]{%
5658 \eqld@decide@select{#3}{#2}{#1}{%
5659 {\eqld@decide@false{\let\eqld@display@depth\@undefined}},%
5660 {{strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
5661 {\relax{\def\eqld@display@depth{#1}}}}}

```

Settings concerning page breaks:

```

5662 \eqld@define@key{equations}{prebreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%

```



```

5663     {{force,4,\eqld@decide@true}}{\eqld@displaybreak@pre4}},%
5664     {{high,3}}{\eqld@displaybreak@pre3}},%
5665     {{med,medium,2}}{\eqld@displaybreak@pre2}},%
5666     {{low,1}}{\eqld@displaybreak@pre1}},%
5667     {{0,\eqld@decide@false}}{\eqld@displaybreak@pre0}},%
5668     {{default,inherit,-1}}{\eqld@displaybreak@pre\m@ne}}}}
5669 \eqld@define@key{equations}{postbreak}[4]{\eqld@decide@select{#3}{#2}{#1}}{%
5670     {{force,4,\eqld@decide@true}}{\eqld@displaybreak@post4}},%
5671     {{high,3}}{\eqld@displaybreak@post3}},%
5672     {{med,medium,2}}{\eqld@displaybreak@post2}},%
5673     {{low,1}}{\eqld@displaybreak@post1}},%
5674     {{0,\eqld@decide@false}}{\eqld@displaybreak@post0}},%
5675     {{default,inherit,-1}}{\eqld@displaybreak@post\m@ne}}}}
5676 \eqld@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
5677     \eqld@decide@select{#3}{#2}{#1}}{%
5678     {{full,4}}{\eqld@displaybreak@inter4}},%
5679     {{high,3}}{\eqld@displaybreak@inter3}},%
5680     {{med,medium,2}}{\eqld@displaybreak@inter2}},%
5681     {{low,1}}{\eqld@displaybreak@inter1}},%
5682     {{0,\eqld@decide@false}}{\eqld@displaybreak@inter\z@}}}}
5683 \eqld@define@key{equations}{prepenalty}{%
5684     \eqld@displaybreak@prepen\numexpr#1\relax}
5685 \eqld@define@key{equations}{postpenalty}{%
5686     \eqld@displaybreak@postpen\numexpr#1\relax}
5687 \eqld@define@key{equations,setup}{interpenalty}{%
5688     \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

5689 \eqld@define@key{control}{vspace}[]{\eqld@vspace@add{#1}}
5690 \eqld@define@key{control}{vspace*}[]{\eqld@vspace@addfixedbefore{#1}}
5691 \eqld@define@key{control}{vspace!}[]{\eqld@vspace@addfixedafter{#1}}
5692 \eqld@define@key{control}{break}[4]{\eqld@displaybreak@level{#1}}
5693 \eqld@define@key{control}{penalty}[]{\eqld@displaybreak@star{#1}}

```

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

```

5694 \eqld@define@key{equations}{noskip}[both]{%
5695     \eqld@decide@abovebelow{#3}{#2}{#1}%
5696     {\def\eqld@skip@force@above{5}}%
5697     {\def\eqld@skip@force@below{5}}}
5698 \eqld@define@key{equations}{short}[above]{%
5699     \eqld@decide@abovebelow{#3}{#2}{#1}%
5700     {\def\eqld@skip@force@above{1}}%
5701     {\def\eqld@skip@force@below{1}}}
5702 \eqld@define@key{equations}{long}[both]{%
5703     \eqld@decide@abovebelow{#3}{#2}{#1}%
5704     {\def\eqld@skip@force@above{0}}%
5705     {\def\eqld@skip@force@below{0}}}
5706 \eqld@define@key{equations}{medskip}[both]{%
5707     \eqld@decide@abovebelow{#3}{#2}{#1}%
5708     {\def\eqld@skip@force@above{6}}%
5709     {\def\eqld@skip@force@below{6}}}
5710 \eqld@define@key{equations}{par}[par]{%
5711     \eqld@decide@select{#3}{#2}{#1}}{%
5712     {{default,\eqld@decide@false}}{\let\eqld@skip@force@leave\undefined}},%
5713     {{cont,hmode}}{\let\eqld@skip@force@leave\z@}},%
5714     {{par,vmode}}{\let\eqld@skip@force@leave\@ne
5715         \ifdefined\eqld@skip@force@below\else

```

```

5716      \def\eq@skip@force@below{3}%
5717      \fi}},%
5718      {\top}{\let\eq@skip@force@leave\tw@
5719      \ifdefined\eq@skip@force@below\else
5720      \def\eq@skip@force@below{4}%
5721      \fi}}}}

```

Specify vertical spacing explicitly:

```

5722 \eq@define@key{equations}{skip}{%
5723   \def\eq@skip@force@above{7}%
5724   \def\eq@skip@custom@above{#1}%
5725   \let\eq@skip@force@below\eq@skip@force@above
5726   \let\eq@skip@custom@below\eq@skip@custom@above}
5727 \eq@define@key{equations}{aboveskip}{%
5728   \def\eq@skip@force@above{7}%
5729   \def\eq@skip@custom@above{#1}}
5730 \eq@define@key{equations}{belowskip}{%
5731   \def\eq@skip@force@below{7}%
5732   \def\eq@skip@custom@below{#1}}
5733 \eq@define@key{equations}{abovespace}{%
5734   \advance\eq@abovespace@glueexpr#1\relax}
5735 \eq@define@key{equations}{belowspace}{%
5736   \advance\eq@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

5737 \eq@define@key{intertext}{skip}{%
5738   \def\eq@skip@force@above{7}%
5739   \def\eq@skip@custom@above{#1}%
5740   \let\eq@skip@force@below\eq@skip@force@above
5741   \let\eq@skip@custom@below\eq@skip@custom@above}
5742 \eq@define@key{intertext}{aboveskip}{%
5743   \def\eq@skip@force@below{7}%
5744   \def\eq@skip@custom@below{#1}}
5745 \eq@define@key{intertext}{belowskip}{%
5746   \def\eq@skip@force@above{7}%
5747   \def\eq@skip@custom@above{#1}}
5748 \eq@define@key{intertext}{noskip}[both]{%
5749   \eq@decide@abovebelow{#3}{#2}{#1}%
5750   {\def\eq@skip@force@below{5}}%
5751   {\def\eq@skip@force@above{5}}}
5752 \eq@define@key{intertext}{short}[both]{%
5753   \eq@decide@abovebelow{#3}{#2}{#1}%
5754   {\def\eq@skip@force@below{1}}%
5755   {\def\eq@skip@force@above{1}}}
5756 \eq@define@key{intertext}{long}[both]{%
5757   \eq@decide@abovebelow{#3}{#2}{#1}%
5758   {\def\eq@skip@force@below{0}}%
5759   {\def\eq@skip@force@above{0}}}
5760 \eq@define@key{intertext}{medskip}[both]{%
5761   \eq@decide@abovebelow{#3}{#2}{#1}%
5762   {\def\eq@skip@force@below{6}}%
5763   {\def\eq@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

5764 \eq@define@key{setup}{skip,longskip}{%
5765   \abovedisplayskip\glueexpr#1\relax
5766   \belowdisplayskip\abovedisplayskip
5767   \def\eq@skip@long@above{#1}%

```

```

5768 \let\eql@skip@long@below\eql@skip@long@above}
5769 \eql@define@key{setup}{aboveskip,abovelongskip}{%
5770 \abovedisplayskip\glueexpr#1\relax
5771 \def\eql@skip@long@above{#1}}
5772 \eql@define@key{setup}{belowskip,belowlongskip}{%
5773 \belowdisplayskip\glueexpr#1\relax
5774 \def\eql@skip@long@below{#1}}
5775 \eql@define@key{setup}{aboveshortskip}{%
5776 \abovedisplayshortskip\glueexpr#1\relax
5777 \def\eql@skip@short@above{#1}}
5778 \eql@define@key{setup}{belowshortskip}{%
5779 \belowdisplayshortskip\glueexpr#1\relax
5780 \def\eql@skip@short@below{#1}}
5781 \eql@define@key{setup}{tagskip}{%
5782 \def\eql@skip@tag@above{#1}%
5783 \let\eql@skip@tag@below\eql@skip@tag@above}
5784 \eql@define@key{setup}{abovetagskip}{%
5785 \def\eql@skip@tag@above{#1}}
5786 \eql@define@key{setup}{belowtagskip}{%
5787 \def\eql@skip@tag@below{#1}}
5788 \eql@define@key{setup}{medskip}{%
5789 \def\eql@skip@med@above{#1}%
5790 \let\eql@skip@med@below\eql@skip@med@above}
5791 \eql@define@key{setup}{abovemedskip}{%
5792 \def\eql@skip@med@above{#1}}
5793 \eql@define@key{setup}{belowmedskip}{%
5794 \def\eql@skip@med@below{#1}}
5795 \eql@define@key{setup}{abovetopskip}{%
5796 \def\eql@skip@top@above{#1}}
5797 \eql@define@key{setup}{belowtopskip}{%
5798 \def\eql@skip@top@below{#1}}
5799 \eql@define@key{setup}{aboveparskip}{%
5800 \def\eql@skip@par@above{#1}}
5801 \eql@define@key{setup}{belowparskip}{%
5802 \def\eql@skip@par@below{#1}}
5803 \eql@define@key{setup}{abovecontskip}{%
5804 \eql@decide@select{#3}{#2}{#1}{%
5805   {{hide}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%
5806   {\relax{\def\eql@skip@cont@above{#1}}}}}
5807 \eql@define@key{setup}{belowcontskip}{%
5808 \def\eql@skip@cont@below{#1}}
5809 \eql@define@key{setup}{shortmode}{%
5810 \eql@decide@select{#3}{#2}{#1}{%
5811   {{\eql@decide@false,never}{\def\eql@skip@mode@short{0}}},%
5812   {{above,neverbelow,belowoff}{\def\eql@skip@mode@short{1}}},%
5813   {{belowone,belowsingle}{\def\eql@skip@mode@short{2}}},%
5814   {{belowall,always,on}{\def\eql@skip@mode@short{3}}}}}
5815 \eql@define@key{setup}{abovecontmode}{%
5816 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}
5817 \eql@define@key{setup}{belowcontmode}{%
5818 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
5819 \eql@define@key{setup}{aboveparmode}{%
5820 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
5821 \eql@define@key{setup}{belowparmode}{%
5822 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
5823 \eql@define@key{setup}{abovetopmode}{%
5824 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
5825 \eql@define@key{setup}{belowtopmode}{%

```

```
5826 \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@below}
```

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

```
5827 \eqld@define@key{equations,subequations}{label}{\eqld@tags@addblock@label{#1}}
5828 \eqld@define@key{equations,subequations}{labelname}{\eqld@tags@addblock@name{#1}}
5829 \eqld@define@key{equations,subequations}{tag}{\eqld@tags@addblock@tag{#1}}
5830 \eqld@define@key{equations,subequations}{tag*}{%
5831 \eqld@tags@addblock@tagform@off\eqld@tags@addblock@tag{#1}}
5832 \eqld@define@key{equations,subequations}{taglabel}{\eqld@tags@addblock@ref{#1}}
```

TODO: describe

```
5833 \eqld@define@key{control}{label}{\eqld@tags@add@label{#1}}
5834 \eqld@define@key{control}{labelname}{\eqld@tags@add@name{#1}}
5835 \eqld@define@key{control}{tag}{\eqld@tags@add@tag{#1}}
5836 \eqld@define@key{control}{tag*}{\eqld@tags@add@tagform@off\eqld@tags@add@tag{#1}}
5837 \eqld@define@key{control}{taglabel}{\eqld@tags@add@ref{#1}}
5838 \eqld@define@key{control}{shifttag}{\eqld@tags@add@raiseshift{#1}}
5839 \eqld@define@key{control}{smashtag}{\eqld@tags@add@raisesmash{#1}}
5840 \eqld@define@key{control}{pushtag}{\eqld@tags@add@forceraise}
```

TODO: describe

```
5841 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
5842 \eqld@define@key{setup}{autolabel}[true]{%
5843 \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}
5844 \eqld@define@key{setup}{autotag}[true]{%
5845 \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}
```

Tag Spacing. Configure horizontal spacing for equation tags:

```
5846 \eqld@define@key{equations,setup}{tagmargin}[auto]{%
5847 \eqld@decide@select{#3}{#2}{#1}{%
5848 {{auto,\eqld@decide@false}{\let\eqld@tagmargin@val\undefined}},%
5849 {\relax{\def\eqld@tagmargin@val{#1}}}}}
5850 \eqld@define@key{equations,setup}{tagmargin*}{%
5851 \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
5852 \eqld@define@key{equations,setup}{tagmarginratio}{%
5853 \eqld@tagmargin@ratio@dimexpr#1pt\relax}
5854 \eqld@define@key{equations,setup}{tagmarginthreshold}{%
5855 \def\eqld@tagmargin@threshold{#1}}
5856 \eqld@define@key{equations,setup}{mintagsep}{\def\eqld@tagsepmin@val{#1}}
5857 \eqld@define@key{equations,setup}{mintagwidth}{%
5858 \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
5859 \eqld@define@key{equations,setup}{mintagwidth*}{%
5860 \settowidth\eqld@tagwidthmin@{#1}}
5861 \eqld@define@key{equations,setup}{tagsnap}{%
5862 \eqld@decide@select{#3}{#2}{#1}{%
5863 {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
5864 {\relax{\def\eqld@tagpos@snap{#1}}}}}
```

Tag Layout. Configure methods to declare equation tag layout:

```
5865 \eqld@define@key{equations,setup}{tagbox,taglayout}{%
5866 \eqld@tags@taglayout@set{#1}}
5867 \eqld@define@key{equations,setup}{tagbox*,taglayout*}{%
5868 \eqld@tags@taglayout@set@direct{#1}}
5869 \eqld@define@key{equations,setup}{tagform}{%}
```

```

5870 \eq@tags@tagform@set#1}
5871 \eq@define@key{equations,setup}{tagform*}{%
5872 \eq@tags@tagform@set@direct{#1}}
5873 \eq@define@key{equations,setup}{subeqtemplate}{%
5874 \eq@subequations@template@set{#1}}

5875 \eq@define@key{control}{tagbox,taglayout}{%
5876 \global\eq@append\eq@tags@container{\eq@tags@taglayout@set{#1}}}
5877 \eq@define@key{control}{tagbox*,taglayout*}{%
5878 \global\eq@append\eq@tags@container{\eq@tags@taglayout@set@direct{#1}}}
5879 \eq@define@key{control}{tagform}{%
5880 \global\eq@append\eq@tags@container{\eq@tags@tagform@set#1}}
5881 \eq@define@key{control}{tagform*}[###1]{%
5882 \global\eq@append\eq@tags@container{\eq@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

5883 \eq@define@key{equations,setup}{numberline,number,num,numline,n}[all]{%
5884 \eq@decide@select{#3}{#2}{#1}{%
5885   {\eq@decide@false,0,*}{\let\eq@numbering@active\eq@false}},%
5886   {\eq@decide@true,!}{\let\eq@numbering@active\eq@true}},%
5887   {none,n,-}{\let\eq@numbering@mode\eq@numbering@mode@multi
5888     \let\eq@numbering@active\eq@false}},%
5889   {single,1}{\let\eq@numbering@mode\eq@numbering@mode@single
5890     \let\eq@numbering@active\eq@true}},%
5891   {multi,0}{\let\eq@numbering@mode\eq@numbering@mode@multi
5892     \let\eq@numbering@active\eq@true}},%
5893   {\relax{\eq@numbering@set{#1}}}}
5894 \eq@define@key{equations,setup}{nonumber,nn,*}[]{%
5895   \let\eq@numbering@active\eq@false}
5896 \eq@define@key{equations,setup}{donumber,dn,!}[]{%
5897   \let\eq@numbering@active\eq@true}
5898 \eq@define@key{equations,setup}{tagsleft,leqno}[]{%
5899   \let\eq@tagsleft\eq@true}
5900 \eq@define@key{equations,setup}{tagsright,reqno}[]{%
5901   \let\eq@tagsleft\eq@false}
5902 \eq@define@key{equations,setup}{tags,eqno}{%
5903   \eq@decide@select{#3}{#2}{#1}{%
5904     {\right,r}{\let\eq@tagsleft\eq@false}},%
5905     {\left,l}{\let\eq@tagsleft\eq@true}}}}
5906 \eq@define@key{equations,setup}{evadetag,avoidtag}[true]{%
5907   \eq@decide@bool{#3}{#2}{#1}\eq@numbering@best@auto}
5908 \eq@define@key{equations,setup}{tagbetween}[true]{%
5909   \eq@decide@bool{#3}{#2}{#1}\eq@tagpos@doconvert}

```

TODO: describe

```

5910 \eq@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
5911 \eq@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
5912 \eq@define@key{control}{numberhere}[]{\eq@numberhere}
5913 \eq@define@key{control}{numbernext}[]{\eq@numbernext}

```

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

```

5914 \eq@define@key{equations,setup}{layout}{\eq@decide@select{#3}{#2}{#1}{%
5915   {\center,c}{\let\eq@layoutleft\eq@false}},%
5916   {\left,l}{\let\eq@layoutleft\eq@true}}}}

```

```

5917 \eqld@define@key{equations,setup}{center}[]{%
5918   \let\eqld@layoutleft\eqld@false}
5919 \eqld@define@key{equations,setup}{flushleft,left}[]{%
5920   \let\eqld@layoutleft\eqld@true}
5921 \eqld@define@key{equations,setup}{leftmargin}{\def\eqld@layoutleftmargin{#1}}
5922 \eqld@define@key{equations,setup}{leftmargin*}{%
5923   \settowidth\dimen@{#1}\edef\eqld@layoutleftmargin{\the\dimen@}}
5924 \eqld@define@key{equations,setup}{minleftmargin}{%
5925   \def\eqld@layoutleftmarginmin{#1}}
5926 \eqld@define@key{equations,setup}{maxleftmargin}{%
5927   \eqld@decide@select{#3}{#2}{#1}{%
5928     {\eqld@decide@false{\def\eqld@layoutleftmarginmax{.5\maxdimen}}},%
5929     {\relax{\def\eqld@layoutleftmarginmax{#1}}}}}

5930 \eqld@define@key{equations,box}{margin}{%
5931   \def\eqld@display@marginleft{#1}\def\eqld@display@marginright{#1}}
5932 \eqld@define@key{equations,box}{marginleft}{\def\eqld@display@marginleft{#1}}
5933 \eqld@define@key{equations,box}{marginright}{\def\eqld@display@marginright{#1}}
5934 \eqld@define@key{equations}{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

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

```

5935 \eqld@define@key{equations,setup}{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5936   {{max,full,4}{\eqld@alignbadness@inf@bad}},%
5937   {{high,3}{\eqld@alignbadness@54\relax}},%
5938   {{med,medium,2}{\eqld@alignbadness@18\relax}},%
5939   {{low,1}{\eqld@alignbadness@6\relax}},%
5940   {{0,\eqld@decide@false}{\eqld@alignbadness@z@}}}}
5941 \eqld@define@key{equations,setup}{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5942   {{max,full,4}{\eqld@tagbadness@inf@bad}},%
5943   {{high,3}{\eqld@tagbadness@54\relax}},%
5944   {{med,medium,2}{\eqld@tagbadness@18\relax}},%
5945   {{low,1}{\eqld@tagbadness@6\relax}},%
5946   {{0,\eqld@decide@false}{\eqld@tagbadness@z@}}}}
5947 \eqld@define@key{equations,setup}{alignbadness}{%
5948   \eqld@alignbadness@numexpr#1\relax}
5949 \eqld@define@key{equations,setup}{tagbadness}{%
5950   \eqld@tagbadness@numexpr#1\relax}
5951 \eqld@define@key{equations,setup}{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5952   {{0,\eqld@decide@false}{\def\eqld@colsepmin@val{0pt}}},%
5953   {\relax{\def\eqld@colsepmin@val{#1}}}}}
5954 \eqld@define@key{equations,setup}{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5955   {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
5956   {\relax{\def\eqld@colsepmax@val{#1}}}}}
5957 \eqld@define@key{equations,setup}{fulllength}[true]{%
5958   \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

```

TODO: is boxcolsep vs breakcolsep okay??!

```

5959 \eqld@define@key{equations,setup}{linesep}{\eqld@decide@select{#3}{#2}{#1}{%
5960   {{0,\eqld@decide@false}{\def\eqld@break@line@sep{0pt}}},%
5961   {\relax{\def\eqld@break@line@sep{#1}}}}}
5962 \eqld@define@key{equations,setup}{linesep*}{\eqld@decide@select{#3}{#2}{#1}{%
5963   {{0,\eqld@decide@false}{\def\eqld@break@line@shortsep{0pt}}},%
5964   {\relax{\def\eqld@break@line@shortsep{#1}}}}}
5965 \eqld@define@key{box,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5966   {{0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
5967   {{short,*}{\def\eqld@box@colsep{\eqld@box@shortsep}}},%

```

```

5968     {\relax{\def\eql@box@colsep{#1}}}%
5969     \let\eql@break@col@sep\eql@box@colsep
5970 \eql@define@key{equations}{colsep}{\eql@decide@select{#3}{#2}{#1}{%
5971     {0,\eql@decide@false}{\def\eql@break@col@sep{0pt}}},%
5972     {\relax{\def\eql@break@col@sep{#1}}}%
5973     \let\eql@colsepmin@val\eql@box@colsep
5974     \let\eql@colsepmax@val\eql@box@colsep
5975     \let\eql@box@colsep\eql@break@col@sep}
5976 \eql@define@key{equations,setup}{colsep*}{\eql@decide@select{#3}{#2}{#1}{%
5977     {0,\eql@decide@false}{\def\eql@break@col@shortsep{0pt}}},%
5978     {\relax{\def\eql@break@col@shortsep{#1}}}%
5979 \eql@define@key{box,setup}{colsep*}{\eql@decide@select{#3}{#2}{#1}{%
5980     {0,\eql@decide@false}{\def\eql@box@shortsep{0pt}}},%
5981     {\relax{\def\eql@box@shortsep{#1}}}%
5982 \eql@define@key{box,setup}{condsep}{\eql@decide@select{#3}{#2}{#1}{%
5983     {0,\eql@decide@false}{\def\eql@box@condsep{0pt}}},%
5984     {{short,*}{\def\eql@box@condsep{\eql@box@shortsep}}},%
5985     {\relax{\def\eql@box@condsep{#1}}}%
5986 \eql@define@key{box,setup}{matrixsep}{\eql@decide@select{#3}{#2}{#1}{%
5987     {0,\eql@decide@false}{\def\eql@box@matrixsep{0pt}}},%
5988     {{short,*}{\def\eql@box@matrixsep{\eql@box@shortsep}}},%
5989     {\relax{\def\eql@box@matrixsep{#1}}}%

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5990 \eql@define@key\eql@keyall{shape}[default]{\eql@shape@set{#1}}
5991 \eql@define@key\eql@keyall{padding,pad}[indent]{%
5992     \eql@decide@select{#3}{#2}{#1}{%
5993         {{max}{\let\eql@paddingleft@val\@undefined}},%
5994         {{indent}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5995         {{0,\eql@decide@false}{\def\eql@paddingleft@val{0pt}}},%
5996         {\relax{\def\eql@paddingleft@val{#1}}}%
5997         \let\eql@paddingright@val\eql@paddingleft@val}
5998 \eql@define@key\eql@keyall{padleft}[indent]{%
5999     \eql@decide@select{#3}{#2}{#1}{%
6000         {{max}{\let\eql@paddingleft@val\@undefined}},%
6001         {{indent}{\def\eql@paddingleft@val{\eql@indent@val}}},%
6002         {{0,\eql@decide@false}{\def\eql@paddingleft@val{0pt}}},%
6003         {\relax{\def\eql@paddingleft@val{#1}}}%
6004 \eql@define@key\eql@keyall{padright}[indent]{%
6005     \eql@decide@select{#3}{#2}{#1}{%
6006         {{max}{\let\eql@paddingright@val\@undefined}},%
6007         {{indent}{\def\eql@paddingright@val{\eql@indent@val}}},%
6008         {{0,\eql@decide@false}{\def\eql@paddingright@val{0pt}}},%
6009         {\relax{\def\eql@paddingright@val{#1}}}%
6010 \eql@define@key\eql@keyall{indent}[2em]{%
6011     \def\eql@indent@val{#1}}

```

TODO: describe

```

6012 \def\eql@shape@rel{\eqncontrol{align=left}}
6013 \def\eql@shape@cont{\eqncontrol{align=left,shift=*}}
6014 \eql@define@key\eql@keyall{shaperel}[]{\def\eql@shape@rel{#1}}
6015 \eql@define@key\eql@keyall{shapecont}[]{\def\eql@shape@cont{#1}}

```

TODO: describe

```

6016 \eql@define@key{control}{align}[]{%
6017     \eql@decide@select{#3}{#2}{#1}{%

```



```

6018    {\l,left}{\global\eql@append\eql@cell@container{\eql@shape@pos@z@}}{,}%
6019    {\c,center}{\global\eql@append\eql@cell@container{\eql@shape@pos@\@ne}}{,}%
6020    {\r,right}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw@}}{}}
6021 \eql@define@key{control}{shift,shiffto}[]{%
6022   \eql@decide@select{#3}{#2}{#1}{%
6023     {\*,indent}{\eql@shape@alignamount@set{\eql@indent@}}{,}%
6024     {\!,outdent}{\eql@shape@alignamount@set{-\eql@indent@}}{,}%
6025     {\relax{\eql@shape@alignamount@set{#1}}}{}}
6026 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

6027 \eql@define@key\eql@keyall{classout}{\eql@class@innerleft@set{#1}}
6028 \eql@define@key\eql@keyall{classin}{\eql@class@innerright@set{#1}}
6029 \eql@define@key\eql@keyall{classlead,classin*}{\eql@class@innerlead@set{#1}}
6030 \eql@define@key\eql@keyall{rel}{\def\eql@class@rel@symb{#1}}
6031 \eql@define@key\eql@keyall{classbreak}{\eql@class@rel@break@set{#1}}
6032 \eql@define@key\eql@keyall{classstart}{\eql@class@rel@start@set{#1}}
6033 \eql@define@key\eql@keyall{classbreakskip}[adjust]{%
6034   \eql@decide@select{#3}{#2}{#1}{%
6035     {\0,\eql@decide@false}{\def\eql@class@rel@break@skip{0mu}}{,}%
6036     {\adjust,*}{\def\eql@class@rel@break@skip{\thickmuskip-\medmuskip}}{,}%
6037     {\relax{\def\eql@class@rel@break@skip{#1}}}{}}
6038 \eql@define@key{control}{rel}{\eql@class@rel@symb}{\eql@class@rel@make{#1}}
6039 \eql@define@key{control}{rel;}[]{\eql@class@rel@make{}}
6040 \eql@define@key{control}{rel*}[]{\eql@class@rel@make{}}
6041 \eql@define@key\eql@keyall{ampeq}[]{\eql@class@ampeq}
6042 \eql@define@key\eql@keyall{eqamp}[]{\eql@class@eqamp}
6043 \eql@define@key\eql@keyall{class}{\eql@decide@select{#3}{#2}{#1}{%
6044   {\ampeq,amprel,eafter,beforerel}\eql@class@ampeq}{,}%
6045   {\eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}

```

Math Styles. Configure math classes at alignment marker:

```

6046 \eql@define@key\eql@keyall{style}[display]{%
6047   \eql@decide@select{#3}{#2}{#1}{%
6048     {\text}{\let\eql@mathstyle\@empty}}{,}%
6049     {\display}{\let\eql@mathstyle\displaystyle}}}}
6050 \eql@define@key{setup}{casesstyle}[display]{%
6051   \eql@decide@select{#3}{#2}{#1}{%
6052     {\eql@decide@false}{\let\eql@cases@mathstyle\eql@false}}{,}%
6053     {\text}{\let\eql@cases@mathstyle\@empty}}{,}%
6054     {\display}{\let\eql@cases@mathstyle\displaystyle}}}}

```

Punctuation. Configure punctuation defaults: **TODO:** describe

```

6055 \def\eql@punct@all#1#2#3#4#5\eql@punct@end{%
6056   \def\eql@tmp{#4}\def\eql@tmpa{1}%
6057   \ifx\eql@tmp\eql@tmpa
6058     \ifnum#5=1111\relax
6059       \eql@punct@set\eql@punct@col{#1}%
6060       \eql@punct@set\eql@punct@line{#2}%
6061       \eql@punct@set\eql@punct@block{#3}%
6062     \else\ifnum#5=111\relax
6063       \eql@punct@set\eql@punct@line{#1}%
6064       \eql@punct@set\eql@punct@block{#2}%
6065     \else\ifnum#5=11\relax

```



```

6066 \eqlopunct@set\eqlopunct@block{#1}%
6067 \else
6068 \eqlopunct@clear
6069 \fi\fi\fi
6070 \else
6071 \eqlopunct@error{Too many arguments to punctall}%
6072 \fi
6073 }

```

TODO: describe

```

6074 \eqlopunct@define@key\eqlopunct@keyall{punctsep}{\,}{\def\eqlopunct@sep{#1}}
6075 \eqlopunct@define@key\eqlopunct@keyall{punctclass}{\mathclose{}}{\def\eqlopunct@class{#1}}
6076 \eqlopunct@define@key\eqlopunct@keyall{punct}{.}{\eqlopunct@set\eqlopunct@block{#1}}
6077 \eqlopunct@define@key\eqlopunct@keyall{punct*}[]{\eqlopunct@set\eqlopunct@block\relax}
6078 \eqlopunct@define@key\eqlopunct@keyall{punctline}{,}{\eqlopunct@set\eqlopunct@line{#1}}
6079 \eqlopunct@define@key\eqlopunct@keyall{punctline*}[]{\eqlopunct@set\eqlopunct@line\relax}
6080 \eqlopunct@define@key\eqlopunct@keyall{punctcol}{,}{\eqlopunct@set\eqlopunct@col{#1}}
6081 \eqlopunct@define@key\eqlopunct@keyall{punctcol*}[]{\eqlopunct@set\eqlopunct@col\relax}
6082 \eqlopunct@define@key\eqlopunct@keyall{punctcases}{,}{\eqlopunct@set\eqlopunct@cases{#1}}
6083 \eqlopunct@define@key\eqlopunct@keyall{punctcases*}[]{\eqlopunct@set\eqlopunct@cases\relax}
6084 \eqlopunct@define@key\eqlopunct@keyall{punctall}{,}{\eqlopunct@all#111111\eqlopunct@end}
6085 \eqlopunct@define@key\eqlopunct@keyall{punctterm}[true]{%
6086 \eqlopunct@decide@bool{#3}{#2}{#1}\eqlopunct@box@punct@term}

6087 \eqlopunct@define@key{control}{punctsep}{\,}{\def\eqlopunct@sep{#1}}
6088 \eqlopunct@define@key{control}{setpunct}{.}{\eqlopunct@set\eqlopunct@next{#1}}
6089 \eqlopunct@define@key{control}{setpunct}{,}{\eqlopunct@set\eqlopunct@next{#1}}
6090 \eqlopunct@define@key{control}{setpunct*}[]{\let\eqlopunct@next\relax}
6091 \eqlopunct@define@key{control}{punct,punctapply}{\relax}{%
6092 \eqlopunct@set\eqlopunct@next{#1}\eqlopunct@apply@top}
6093 \eqlopunct@define@key{control}{punctline}[]{\eqlopunct@print@line}
6094 \eqlopunct@define@key{control}{punctcol}[]{\eqlopunct@print@col}

```

Frames. **TODO:** describe

```

6095 \eqlopunct@define@key{box}{frame}[\fbox]{%
6096 \def\eqlopunct@box@frame{#1}%
6097 \ifx\eqlopunct@box@frame\empty\let\eqlopunct@box@frame\@firstofone\fi}
6098 \eqlopunct@define@key{box}{wrap}{\eqlopunct@box@wrap{#1}}
6099 \eqlopunct@define@key{box}{ldelim}[r]{\eqlopunct@decide@delim{#3}{#2}{#1}}
6100 \eqlopunct@define@key{box}{ldelim}{\eqlopunct@box@ldelim{#1}}
6101 \eqlopunct@define@key{box}{rdelim}{\eqlopunct@box@rdelim{#1}}
6102 \eqlopunct@define@key{box}{lbrace}[]{\eqlopunct@box@ldelim\lbrace}
6103 \eqlopunct@define@key{box}{rbrace}[]{\eqlopunct@box@rdelim\rbrace}
6104 \eqlopunct@define@key{box}{lbrace,rbrace}[]{\eqlopunct@box@delim\lbrace\rbrace}
6105 \eqlopunct@define@key{box}{braces}[lr]{%
6106 \eqlopunct@decide@select{#3}{#2}{#1}{%
6107 {\eqlopunct@decide@false}{\eqlopunct@box@wrap}{}}},%
6108 {\l,left}{\eqlopunct@box@ldelim\lbrace}},%
6109 {\r,right}{\eqlopunct@box@rdelim\rbrace}},%
6110 {\eqlopunct@decide@true,lr,both}{\eqlopunct@box@delim\lbrace\rbrace}}}}

```

TODO: describe

```

6111 \eqlopunct@define@key{control}{framecell}[\fbox]{%
6112 \global\eqlopunct@append\eqlopunct@cell@container{\def\eqlopunct@frame@cmd{#1}}}
6113 \eqlopunct@define@key{control}{frametag}[\fbox]{%
6114 \global\eqlopunct@append\eqlopunct@tags@container{\def\eqlopunct@tags@frame@cmd{#1}}}

```

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

```
6115 \eqld@define@key{equations,box}{alt}{}
```

Injections.

```
6116 \eqld@define@key{control}{inject}{%
6117   \global\eqld@append\eqld@interline@container{%
6118     \eqld@append\eqld@display@injectbefore{#1}}
6119 \eqld@define@key{control}{inject*}{%
6120   \global\eqld@append\eqld@interline@container{%
6121     \eqld@append\eqld@display@injectafter{#1}}
6122 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
6123 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
6124 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
6125 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}
```

TODO: describe

```
6126 \eqld@define@key{markline}{pos}{%
6127   \eqld@decide@select{#3}{#2}{#1}{%
6128     {{below,push}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
6129     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
6130     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}%
6131 \eqld@define@key{markline}{below,push}[]{%
6132   \let\eqld@markline@pos\eqld@markline@pos@below}
6133 \eqld@define@key{markline}{baseline}[]{%
6134   \let\eqld@markline@pos\eqld@markline@pos@baseline}
6135 \eqld@define@key{markline}{bottom}[]{%
6136   \let\eqld@markline@pos\eqld@markline@pos@bottom}
6137 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
6138 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}
6139 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
6140 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}
6141 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
6142 \eqld@define@key{setup}{markpos}{%
6143   \eqld@decide@select{#3}{#2}{#1}{%
6144     {{below}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
6145     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
6146     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}%

```

Global Switches. Set global switches:

```
6147 \let\eqld@multi@linesfallback\eqld@false
6148 \let\eqld@scan@par\eqld@false
6149 \let\eqld@single@cr@mode\eqld@false
6150 \let\eqld@amp@mode\eqld@true
6151 \let\eqld@ampproof@active\eqld@false
6152 \let\eqld@parseopt@warn@env\eqld@warn@parseopt
6153 \let\eqld@parseopt@warn@cr\@empty

6154 \eqld@define@key{equations,setup}{linesfallback}[true]{%
6155   \eqld@decide@select{#3}{#2}{#1}{%
6156     {\eqld@decide@false{\let\eqld@multi@linesfallback\eqld@false}},%
6157     {\reuse,lean}{\let\eqld@multi@linesfallback\z@}},%
6158     {\measure,full,\eqld@decide@true}{\let\eqld@multi@linesfallback\eqld@true}}}%
6159 \eqld@define@key{setup}{ampproof}[true]{%
6160   \eqld@decide@bool{#3}{#2}{#1}\eqld@ampproof@active}
```

```

6161 \eqld@define@key{equations,setup}{equationcr}[true]{%
6162   \eqld@decide@select{#3}{#2}{#1}{%
6163     {\eqld@decide@false{\let\eqld@single@cr@mode\eqld@false}},%
6164     {\eqld@decide@true,break}{\let\eqld@single@cr@mode\eqld@break@cr}},%
6165     {\error,verbose}{\let\eqld@single@cr@mode\eqld@single@cr@error}}}%
6166 \eqld@define@key\eqld@keyall{amp}[true]{%
6167   \eqld@decide@bool{#3}{#2}{#1}{\eqld@amp@mode}}
6168 \eqld@define@key\eqld@keyall{rescan}[true]{%
6169   \eqld@decide@if{#3}{#2}{#1}{%
6170     {\let\eqld@scan@body\eqld@scan@body@rescan}%
6171     {\let\eqld@scan@body\eqld@scan@body@dump}}
6172 \eqld@define@key\eqld@keyall{scanpar}[true]{%
6173   \eqld@decide@bool{#3}{#2}{#1}{\eqld@scan@par}}
6174 \eqld@define@key{setup}{defaults}{%
6175   \eqld@decide@select{#3}{#2}{#1}{%
6176     {\classic}{\eqld@defaults@classic}},%
6177     {\eqnlines}{\eqld@defaults@eqnlines}}}%
6178 \eqld@define@key\eqld@keyall{verbose}[true]{%
6179   \eqld@decide@if{#3}{#2}{#1}{\eqld@verbose@on\eqld@verbose@off}}

```

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

```

6180 \let\eqld@provide@opt@env\tw@
6181 \let\eqld@provide@opt@amsmathpatch\eqld@false
6182 \let\eqld@provide@opt@backup\eqld@false
6183 \let\eqld@provide@opt@ang\eqld@true
6184 \let\eqld@provide@opt@eqref\eqld@true
6185 \let\eqld@provide@opt@matrix\eqld@true

6186 \eqld@define@key{setup}{amsmathends,amsmathpatch}[true]{%
6187   \eqld@error@packageoption{#2}%
6188   \eqld@decide@bool{#3}{#2}{#1}{\eqld@provide@opt@amsmathpatch}}
6189 \eqld@define@key{setup}{backup}[true]{%
6190   \eqld@error@packageoption{#2}%
6191   \eqld@decide@bool{#3}{#2}{#1}{\eqld@provide@opt@backup}}
6192 \eqld@define@key{setup}{env}[equation]{%
6193   \eqld@error@packageoption{#2}%
6194   \eqld@decide@select{#3}{#2}{#1}{%
6195     {\none,\eqld@decide@false}{\let\eqld@provide@opt@env\z@}},%
6196     {\equation,latex}{\let\eqld@provide@opt@env\ne}},%
6197     {\amsmath,all,\eqld@decide@true}{\let\eqld@provide@opt@env\tw@}}}%
6198 \eqld@define@key{setup}{ang}[true]{%
6199   \eqld@error@packageoption{#2}%
6200   \eqld@decide@bool{#3}{#2}{#1}{\eqld@provide@opt@ang}}
6201 \eqld@define@key{setup}{eqref}[true]{%
6202   \eqld@error@packageoption{#2}%
6203   \eqld@decide@bool{#3}{#2}{#1}{\eqld@provide@opt@eqref}}
6204 \eqld@define@key{setup}{matrix}[true]{%
6205   \eqld@error@packageoption{#2}%
6206   \eqld@decide@bool{#3}{#2}{#1}{\eqld@provide@opt@matrix}}

```

Shortcut Options. **TODO:** describe

```

6207 \def\eqld@parseopt@nonumber#1{\eqnaddopt{nonumber}\eqld@parseopt@peek}
6208 \def\eqld@parseopt@donumber#1{\eqnaddopt{donumber}\eqld@parseopt@peek}
6209 \def\eqld@parseopt@single#1{\eqnaddopt{single}\eqld@parseopt@peek}
6210 \def\eqld@parseopt@lines#1{\eqnaddopt{lines}\eqld@parseopt@peek}
6211 \def\eqld@parseopt@eqamp#1{\eqnaddopt{eqamp}\eqld@parseopt@peek}

```

```

6212 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
6213 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
6214 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
6215 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
6216 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
6217 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
6218 \def\eql@parseopt@punctclear#1{\eql@parseopt@peek'\sim}
6219 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
6220 \def\eql@parseopt@punctnext#1#2{%
6221   \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
6222 \def\eql@parseopt@punctblock#1#2{%
6223   \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
6224 \def\eql@parseopt@punctterm#1{\let\eql@punct@term\eql@true\eql@parseopt@peek}
6225 \def\eql@parseopt@relsymsb#1{\eql@parseopt@peek?\eql@class@rel@symb}
6226 \def\eql@parseopt@relcont#1{\eql@parseopt@peek?{}}
6227 \def\eql@parseopt@relstart#1{\eql@parseopt@peek?\relax}
6228 \def\eql@parseopt@relord#1{\eql@parseopt@peek?{}}
6229 \def\eql@parseopt@vspace[#1]{%
6230   \advance\eql@vspaceskip@\glueexpr#1\relax\eql@parseopt@peek}

```

16.3 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:

```

6231 \def\eql@defaults@classic{%
6232   \eqnlineset{numberline=all}%
6233   \eqnlineset{condsep={1em}}%
6234   \eqnlineset{matrixsep={2\arraycolsep}}%
6235   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
6236   \eqnlineset{maxcolsep=off}%
6237   \eqnlineset{spread={\jot}}%
6238   \eqnlineset{tagmargin}%
6239   \eqnlineset{tagmarginratio=1}%
6240   \eqnlineset{tagmarginthreshold=0.5}%
6241   \eqnlineset{leftmargin={\leftmargini}}%
6242   \eqnlineset{padding=max}%
6243   \eqnlineset{evadetag=off}%
6244   \eqnlineset{displayheight=off}%
6245   \eqnlineset{displaydepth=off}%
6246   \eqnlineset{shortmode=belowsingle}%
6247   \eqnlineset{abovecontmode=short}%
6248   \eqnlineset{belowcontmode=short}%
6249   \eqnlineset{aboveparmode=long}%
6250   \eqnlineset{belowparmode=long}%
6251   \eqnlineset{abovetopmode=long}%
6252   \eqnlineset{belowtopmode=long}%
6253   \eqnlineset{abovelongskip={\abovedisplayskip}}%
6254   \eqnlineset{belowlongskip={\belowdisplayskip}}%
6255   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
6256   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
6257   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
6258   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%

```

```

6259 \eqnlineset{abovecontskip=0pt}%
6260 \eqnlineset{belowcontskip=0pt}%
6261 \eqnlineset{aboveparskip=0pt}%
6262 \eqnlineset{belowparskip=0pt}%
6263 \eqnlineset{abovetopskip=0pt}%
6264 \eqnlineset{belowtopskip=0pt}%
6265 \eqnlineset{abovetagskip=0pt}%
6266 \eqnlineset{belowtagskip=0pt}%
6267 \eqnlineset{allowbreaks=0}%
6268 \eqnlineset{equationcr=off}%
6269 \eqnlineset{amp=off}%
6270 \eqnlineset{modifier=off}%
6271 \eqnlineset{linesfallback=false}%
6272 \eqnlineset{casesstyle=text}%
6273 \eqnlineset{sqropt=nonumber}%
6274 \eqnlineset{angopt=nonumber}%
6275 }

```

values based on 10pt vs 12pt

`\eqnlines` The (default) preset `\eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the classic spacing for a 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):

```

6276 \def\eqn@defaults@eqnlines{%
6277   \eqnlineset{numberline=all}%
6278   \eqnlineset{condsep=short}%
6279   \eqnlineset{matrixsep=short}%
6280   \eqnlineset{mintagsep=.5em}%
6281   \eqnlineset{maxcolsep=2em}%
6282   \eqnlineset{spread={0.2\normalbaselineskip}}%
6283   \eqnlineset{tagmargin}%
6284   \eqnlineset{tagmarginratio=.334}%
6285   \eqnlineset{tagmarginthreshold=0.5}%
6286   \eqnlineset{leftmargin={\leftmargini}}%
6287   \eqnlineset{padding=0pt}%
6288   \eqnlineset{evadetag}%
6289   \eqnlineset{displayheight=strut}%
6290   \eqnlineset{displaydepth=strut}%
6291   \eqnlineset{shortmode=above}%
6292   \eqnlineset{abovecontmode=noskip}%
6293   \eqnlineset{belowcontmode=long}%
6294   \eqnlineset{aboveparmode=long}%
6295   \eqnlineset{belowparmode=long}%
6296   \eqnlineset{abovetopmode=noskip}%
6297   \eqnlineset{belowtopmode=long}%
6298   \eqnlineset{longskip={0.75\normalbaselineskip
6299     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
6300   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
6301     plus 0.25\normalbaselineskip}}%
6302   \eqnlineset{belowshortskip={0.0\normalbaselineskip
6303     plus 0.25\normalbaselineskip}}%
6304   \eqnlineset{medskip={0.4\normalbaselineskip
6305     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
6306   \eqnlineset{abovecontskip=0pt}%
6307   \eqnlineset{belowcontskip=0pt}%

```

```

6308 \eqnlineset{aboveparskip=0pt}%
6309 \eqnlineset{belowparskip=0pt}%
6310 \eqnlineset{abovetopskip=0pt}%
6311 \eqnlineset{belowtopskip=0pt}%
6312 \eqnlineset{abovetagskip={0.25\normalbaselineskip
6313   minus 0.25\normalbaselineskip}}}%
6314 \eqnlineset{belowtagskip={0.25\normalbaselineskip
6315   minus 0.25\normalbaselineskip}}}%
6316 \eqnlineset{allowbreaks=3}%
6317 \eqnlineset{equationcr=break}%
6318 \eqnlineset{amp=on}%
6319 \eqnlineset{modifier=on,modifierenv=off}%
6320 \eqnlineset{linesfallback=true}%
6321 \eqnlineset{casesstyle=false}%
6322 \eqnlineset{sqropt}%
6323 \eqnlineset{angopt}%
6324 }

```

16.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of `LATEX` and `amsmath` carefully.

Tools.

`\eql@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```

\eql@provide@moveenv
@provide@undefinecmd
@provide@undefineenv
6325 \def\eql@provide@movecmd#1#2{%
6326   \eql@letcs{#1\expandafter}\csname#2\endcsname
6327 }
6328 \def\eql@provide@moveenv#1#2{%
6329   \eql@provide@movecmd{#1}{#2}%
6330   \ifcsname end#2\endcsname
6331     \eql@provide@movecmd{end#1}{end#2}%
6332   \fi
6333 }
6334 \def\eql@provide@undefinecmd#1{%
6335   \eql@letcs{#1}\@undefined
6336 }
6337 \def\eql@provide@undefineenv#1{%
6338   \eql@provide@undefinecmd{#1}%
6339   \eql@provide@undefinecmd{end#1}%
6340 }

```

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:

```

6341 \def\eql@amsmath@endfix#1#2{%

```

```

6342 \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
6343 \expandafter\ifx\csname end#1\endcsname\eql@tmpa
6344 \eql@provide@movecmd{end#1}{end#2}%
6345 \fi
6346 }

```

ql@amsmath@fixmatrix **TODO:** describe

amsmath@fixmatrixend

```

6347 \def\eql@amsmath@fixmatrix#1{%
6348 \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
6349 \begingroup
6350 \let\matrix@check\@gobble
6351 \def\env@matrix{\noexpand\env@matrix}%
6352 \def\env@cases{\noexpand\env@cases}%
6353 \global\edef\eql@tmp{\eql@tmp}%
6354 \endgroup
6355 \eql@letcs{#1}\eql@tmp
6356 }
6357 \def\eql@amsmath@fixmatrixend#1{%
6358 \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
6359 \begingroup
6360 \expandafter\def\expandafter\endmatrix\expandafter{%
6361 \expandafter\unexpanded\expandafter{\endmatrix}}%
6362 \global\long\edef\eql@tmp{\eql@tmp}%
6363 \endgroup
6364 \eql@letcs{end#1}\eql@tmp
6365 }

```

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

```

6366 \def\eql@amsmath@fixends{%
6367 \eql@amsmath@after{%
6368 \eql@amsmath@endfix{flalign}{align}%
6369 \eql@amsmath@endfix{alignat}{align}%
6370 \eql@amsmath@endfix{xalignat}{align}%
6371 \eql@amsmath@endfix{xxalignat}{align}%
6372 \eql@amsmath@endfix{gather*}{gather}%
6373 \eql@amsmath@endfix{multline*}{multline}%
6374 \eql@amsmath@endfix{align*}{align}%
6375 \eql@amsmath@endfix{flalign*}{align}%
6376 \eql@amsmath@endfix{alignat*}{align}%
6377 \eql@amsmath@endfix{xalignat*}{align}%
6378 \eql@amsmath@endfix{gathered}{aligned}%
6379 \eql@amsmath@endfix{alignedat}{aligned}%
6380 }
6381 }

```

@amsmath@fixmatrices Perform the replacement for all amsmath environments whenever amsmath is loaded:

```

6382 \def\eql@amsmath@fixmatrices{%
6383 \eql@amsmath@after{%
6384 \eql@amsmath@fixmatrix{cases}%
6385 \eql@amsmath@fixmatrix{matrix}%
6386 \eql@amsmath@fixmatrix{pmatrix}%
6387 \eql@amsmath@fixmatrixend{pmatrix}%
6388 \eql@amsmath@fixmatrixend{bmatrix}%
6389 \eql@amsmath@fixmatrixend{Bmatrix}%
6390 \eql@amsmath@fixmatrixend{vmatrix}%
6391 \eql@amsmath@fixmatrixend{Vmatrix}%

```

```
6392 }
6393 }
```

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

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

```
6394 \def\eql@provide@backup@amsenv#1{%
6395   \eql@amsmath@after{%
6396     \eql@provide@moveenv{ams#1}{#1}%
6397     \eql@tagging@register@luamml{ams#1}%
6398     \eql@markline@amsthm@move{ams#1}{#1}%
6399   }%
6400 }
```

`provide@backup@amsbox` **TODO:** describe

```
6401 \def\eql@provide@backup@amsbox#1{%
6402   \eql@amsmath@after{%
6403     \eql@provide@moveenv{ams#1}{#1}%
6404   }%
6405 }
```

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

```
6406 \def\eql@provide@backup@eqref{%
6407   \eql@amsmath@after{%
6408     \eql@provide@movecmd{amseqref}{eqref}%
6409   }%
6410 }
```

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

```
6411 \def\eql@provide@backup@multlined{%
6412   \AddToHook{package/mathtools/after}{%
6413     \eql@provide@moveenv{amsmultlined}{multlined}}%
6414 }
```

`provide@backup@equation` The \LaTeX environment `equation` is overwritten by several packages to implement their adjustments. Here we cater for adjustments through `amsmath`, `hyperref` and the PDF tagging mechanism. Copy `equation` and `equation*` whenever `amsmath` is loaded. Whenever `hyperref` is loaded, and `amsmath` is not yet present, backup the original \LaTeX and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original \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:

```
6415 \def\eql@provide@backup@equation{%
6416   \eql@amsmath@after{%
6417     \eql@provide@moveenv{amsequation}{equation}%
6418     \eql@provide@moveenv{amsequation*}{equation*}%
6419     \eql@tagging@register@env{amsequation}%
6420     \eql@tagging@register@env{amsequation*}%
6421     \eql@tagging@register@luamml{amsequation}%

```



```

6422 \eql@tagging@register@luamml{amsequation*}%
6423 \eql@markline@amsthm@move{amsequation}{equation}%
6424 \eql@markline@amsthm@move{amsequation*}{equation*}%
6425 }%
6426 \AddToHook{package/hyperref/after}{%
6427 \@ifpackageloaded{amsmath}{}%
6428 \eql@provide@moveenv{hyperrefequation}{equation}%
6429 \eql@tagging@register@env{hyperrefequation}%
6430 \eql@tagging@register@luamml{hyperrefequation}%
6431 \eql@markline@amsthm@move{hyperrefequation}{equation}%
6432 }%
6433 }%
6434 \@ifpackageloaded{amsmath}{}%
6435 \@ifpackageloaded{hyperref}{%
6436 \let\latexequation\H@equation
6437 \let\endlatexequation\H@endequation
6438 }{\eql@provide@moveenv{latexequation}{equation}}%
6439 \eql@tagging@register@env{latexequation}%
6440 \eql@tagging@register@luamml{latexequation}%
6441 \eql@markline@amsthm@move{latexequation}{equation}%
6442 }%
6443 }

```

e@backup@displaymath **TODO:** describe

```

6444 \def\eql@provide@backup@displaymath{%
6445 \eql@provide@moveenv{latexdisplaymath}{displaymath}%
6446 \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
6447 }

```

@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:

```

6448 \def\eql@provide@backup@subequations{%
6449 \eql@amsmath@after{%
6450 \eql@provide@moveenv{amssubequations}{subequations}%
6451 }%
6452 \AddToHook{package/hyperref/after}{%
6453 \AddToHook{cmd/amssubequations/before}{%
6454 {%
6455 \stepcounter{equation}%
6456 \protected@edef\theHparentequation{\theHequation}%
6457 \addtocounter{equation}{-1}%
6458 }%
6459 \AddToHook{cmd/amssubequations/after}{%
6460 {%
6461 \def\theHequation{\theHparentequation\alph{equation}}%
6462 \ignorespaces
6463 }%
6464 }%
6465 }

```

\eql@provide@backup Backup all amsmath environments:

```

6466 \def\eql@provide@backup{%
6467 \eql@provide@backup@eqref
6468 \eql@provide@backup@equation
6469 \eql@provide@backup@displaymath

```

```

6470 \eql@provide@backup@amsenv{gather}%
6471 \eql@provide@backup@amsenv{multline}%
6472 \eql@provide@backup@amsenv{align}%
6473 \eql@provide@backup@amsenv{flalign}%
6474 \eql@provide@backup@amsenv{alignat}%
6475 \eql@provide@backup@amsenv{xalignat}%
6476 \eql@provide@backup@amsenv{xxalignat}%
6477 \eql@provide@backup@amsenv{gather*}%
6478 \eql@provide@backup@amsenv{multline*}%
6479 \eql@provide@backup@amsenv{align*}%
6480 \eql@provide@backup@amsenv{flalign*}%
6481 \eql@provide@backup@amsenv{alignat*}%
6482 \eql@provide@backup@amsenv{xalignat*}%
6483 \eql@provide@backup@amsbox{gathered}%
6484 \eql@provide@backup@multlined
6485 \eql@provide@backup@amsbox{aligned}%
6486 \eql@provide@backup@amsbox{alignedat}%
6487 \eql@provide@backup@amsbox{cases}%
6488 \eql@provide@backup@amsbox{matrix}%
6489 \eql@provide@backup@amsbox{pmatrix}%
6490 \eql@provide@backup@amsbox{bmatrix}%
6491 \eql@provide@backup@amsbox{Bmatrix}%
6492 \eql@provide@backup@amsbox{vmatrix}%
6493 \eql@provide@backup@amsbox{Vmatrix}%
6494 \eql@provide@backup@subequations
6495 }

```

Replacement amsmath Environments. **TODO:** describe

```

6496 \def\eql@alignat@gobblecol#1{%
6497   \eql@ifnextchar@tight\bgroup{\@firstoftwo{#1}}{#1}}

```

`eql@gathered` (*env.*) Define replacement versions for boxed environments `gathered`, `multlined` and `aligned`
`eql@multlined` (*env.*) which forward to `equationsbox` with specific presets:

`eql@aligned` (*env.*)

```

6498 \newenvironment{eql@gathered}%
6499   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
6500 \newenvironment{eql@multlined}%
6501   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
6502 \newenvironment{eql@aligned}%
6503   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
6504 \newenvironment{eql@alignedat}%
6505   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}%
6506   {\endequationsbox}
6507 \newenvironment{eql@cases}%
6508   {\eqnaddopt{cases}\equationsbox}{\endequationsbox}
6509 \newenvironment{eql@matrix}%
6510   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=.}\equationsbox}%
6511   {\endequationsbox}
6512 \newenvironment{eql@pmatrix}%
6513   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=r}\equationsbox}%
6514   {\endequationsbox}
6515 \newenvironment{eql@bmatrix}%
6516   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=s}\equationsbox}%
6517   {\endequationsbox}
6518 \newenvironment{eql@Bmatrix}%
6519   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=c}\equationsbox}%
6520   {\endequationsbox}

```

```

6521 \newenvironment{eql@vmatrix}%
6522   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=v}\equationsbox}%
6523   {\endequationsbox}
6524 \newenvironment{eql@Vmatrix}%
6525   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=d}\equationsbox}%
6526   {\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.*)

```

6527 \newenvironment{eql@equation}%
6528   {\eqnaddopt{equation,donumber}\equations}\endequations}
6529 \newenvironment{eql@equation*}%
6530   {\eqnaddopt{equation,nonumber}\equations}\endequations}
6531 \newenvironment{eql@displaymath}%
6532   {\eqnaddopt{equation,nonumber}\equations}\endequations}
6533 \newenvironment{eql@gather}%
6534   {\eqnaddopt{lines,donumber}\equations}\endequations}
6535 \newenvironment{eql@gather*}%
6536   {\eqnaddopt{lines,nonumber}\equations}\endequations}
6537 \newenvironment{eql@multline}%
6538   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,donumber}%
6539    \equations}\endequations}
6540 \newenvironment{eql@multline*}%
6541   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,nonumber}%
6542    \equations}\endequations}
6543 \newenvironment{eql@align}%
6544   {\eqnaddopt{columns,donumber}\equations}\endequations}
6545 \newenvironment{eql@align*}%
6546   {\eqnaddopt{columns,nonumber}\equations}\endequations}
6547 \newenvironment{eql@flalign}%
6548   {\eqnaddopt{columns,fulllength,donumber}\equations}\endequations}
6549 \newenvironment{eql@flalign*}%
6550   {\eqnaddopt{columns,fulllength,nonumber}\equations}\endequations}
6551 \newenvironment{eql@alignat}%
6552   {\eqnaddopt{columns,colsep=off,donumber}%
6553    \eql@alignat@gobblecol\equations}\endequations}
6554 \newenvironment{eql@xalignat}%
6555   {\eqnaddopt{columns,donumber}%
6556    \eql@alignat@gobblecol\equations}\endequations}
6557 \newenvironment{eql@xxalignat}%
6558   {\eqnaddopt{columns,fulllength,donumber}%
6559    \eql@alignat@gobblecol\equations}\endequations}
6560 \newenvironment{eql@alignat*}%
6561   {\eqnaddopt{columns,colsep=off,nonumber}%
6562    \eql@alignat@gobblecol\equations}\endequations}
6563 \newenvironment{eql@xalignat*}%
6564   {\eqnaddopt{columns,nonumber}%
6565    \eql@alignat@gobblecol\equations}\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):

```

6566 \def\eql@provide@onlyonce#1#2{%
6567   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
6568   \ifx\eql@tmp\eql@tmpa
6569     \let\eql@tmp\@empty
6570   \fi
6571   \ifx\eql@tmp\@empty
6572     \let\eql@tmp\@undefined
6573     \ifx\@nodocument\relax
6574       \def\eql@tmp{#1}%
6575     \fi
6576     \ifcsname eql@provided@#1\endcsname
6577       \def\eql@tmp{#1}%
6578     \fi
6579     \eql@letcs{eql@provided@#1}\eql@true
6580   \else
6581     \def\eql@tmpa{*}%
6582     \ifx\eql@tmp\eql@tmpa
6583       \def\eql@tmp{#1}%
6584     \fi
6585   \fi
6586 }
```

`\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:

```

6587 \def\eql@provide@eqref#1{%
6588   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
6589   \ifx\eql@tmp\eql@tmpa
6590     \let\eql@tmp\@empty
6591   \fi
6592   \ifx\eql@tmp\@empty
6593     \eql@provide@onlyonce{eqref}{}%
6594   \else
6595     \def\eql@tmpa{*}%
6596     \ifx\eql@tmp\eql@tmpa
6597       \def\eql@tmp{eqref}%
6598     \else
6599       \edef\eql@tmp{\expandafter\@gobble\string#1}%
6600     \fi
6601   \fi
6602   \ifdefined\eql@tmp
6603     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
6604   \else
6605     \eql@amsmath@after{%
6606       \eql@provide@movecmd{eqref}{eql@eqref}%
6607     }%
6608     \AddToHook{package/mathtools/after}{%
6609       \eql@provide@movecmd{eqref}{eql@eqref}%
6610     }%
6611     \eql@provide@movecmd{eqref}{eql@eqref}%
6612     \eql@amsmath@undefine\eqref
6613   \fi
6614 }
```

`\eql@provide@amsenv` Provide one of the amsmath environments. Copy into place, and copy again when amsmath is loaded. Remove definition before amsmath is loaded in the future to avoid an error:

```

6615 \def\eql@provide@amsenv#1#2{%
6616   \eql@provide@onlyonce{#1}{#2}%
6617   \ifdefined\eql@tmp
6618     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6619     \eql@tagging@register@luamml{\eql@tmp}%
6620     \eql@markline@amsthm@register{\eql@tmp}%
6621   \else
6622     \eql@amsmath@after{%
6623       \eql@provide@moveenv{#1}{eql@#1}%
6624       \eql@markline@amsthm@register{#1}%
6625     }%
6626     \AddToHook{package/mathtools/after}{%
6627       \eql@provide@moveenv{#1}{eql@#1}%
6628       \eql@markline@amsthm@register{#1}%
6629     }%
6630     \eql@provide@moveenv{#1}{eql@#1}%
6631     \eql@markline@amsthm@register{#1}%
6632     \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6633   \fi
6634 }
```

`\eql@provide@amsbox` Provide one of the amsmath subequation structures. Copy into place, and copy again when amsmath is loaded. Remove definition before amsmath is loaded in the future to avoid an error:

```

6635 \def\eql@provide@amsbox#1#2{%
6636   \eql@provide@onlyonce{#1}{#2}%
6637   \ifdefined\eql@tmp
6638     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6639   \else
6640     \eql@amsmath@after{%
6641       \eql@provide@moveenv{#1}{eql@#1}%
6642       \AddToHook{package/mathtools/after}{%
6643         \eql@provide@moveenv{#1}{eql@#1}%
6644         \eql@provide@moveenv{#1}{eql@#1}%
6645       \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6646     \fi
6647 }
```

`\eql@provide@multlined` Provide mathtools environment multlined. Copy into place, and copy again when mathtools is loaded. Remove definition before mathtools is loaded in the future to avoid an error:

```

6648 \def\eql@provide@multlined#1{%
6649   \eql@provide@onlyonce{multlined}{#1}%
6650   \ifdefined\eql@tmp
6651     \eql@provide@moveenv{\eql@tmp}{eql@multlined}%
6652   \else
6653     \AddToHook{package/mathtools/after}{%
6654       \eql@provide@moveenv{multlined}{eql@multlined}%
6655       \eql@provide@moveenv{multlined}{eql@multlined}%
6656       \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
6657         \eql@provide@undefineenv{multlined}}}%
6658     \fi
6659 }
```

`\eql@provide@matrix` Provide the cases and matrix environments. Copy into place, and copy again when `amsmath` is loaded:

```

6660 \def\eql@provide@matrix#1#2#3{%
6661   \eql@provide@onlyonce{#1}{#3}%
6662   \ifdefined\eql@tmp
6663     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6664     \eql@tagging@register@luamml{\eql@tmp}%
6665   \else
6666     \eql@amsmath@after{%
6667       \eql@provide@moveenv{#1}{eql@#1}%
6668     }%
6669     \eql@provide@moveenv{#1}{eql@#1}%
6670     \ifdefined#2\eql@amsmath@before{\eql@provide@undefineenv{#1}}\fi%
6671   \fi
6672 }
```

`\eql@provide@equation` Provide the environment `equation`. Copy into place, and copy again when `amsmath` or `hyperref` are loaded. When PDF tagging is active, the environment is modified at `\begin{document}` in an undesirable fashion, so copy the definition again:

```

6673 \def\eql@provide@equation#1{%
6674   \eql@provide@onlyonce{equation}{#1}%
6675   \ifdefined\eql@tmp
6676     \eql@provide@moveenv{\eql@tmp}{eql@equation}%
6677     \eql@tagging@register@luamml{\eql@tmp}%
6678     \eql@markline@amsthm@register{\eql@tmp}%
6679   \else
6680     \eql@amsmath@after{%
6681       \eql@provide@moveenv{equation}{eql@equation}%
6682       \eql@markline@amsthm@register{equation}%
6683     }%
6684     \AddToHook{package/hyperref/after}{%
6685       \ifpackageloaded{amsmath}{%
6686         \eql@provide@moveenv{equation}{eql@equation}%
6687         \eql@markline@amsthm@register{equation}%
6688       }%
6689     }%
6690     \eql@provide@moveenv{equation}{eql@equation}%
6691     \eql@markline@amsthm@register{equation}%
6692     \ifdefined\eql@tagging@on
6693       \AddToHook{begindocument/end}{%
6694         \eql@provide@moveenv{equation}{eql@equation}%
6695         \eql@markline@amsthm@register{equation}%
6696       }%
6697     \fi
6698   \fi
6699 }
```

`\provide@equationstar` Provide the environment `equation*`. 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:

```

6700 \def\eql@provide@equationstar#1{%
6701   \eql@provide@onlyonce{equation*}{#1}%
6702   \ifdefined\eql@tmp
6703     \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
6704     \eql@tagging@register@luamml{\eql@tmp}%

```

```

6705 \eq\markline@amsthm@register{\eq\@tmp}%
6706 \else
6707 \eq\amsmath@after{%
6708 \eq\provide@moveenv{equation*}{\eq\@equation*}%
6709 \eq\markline@amsthm@register{equation*}%
6710 }%
6711 \eq\provide@moveenv{equation*}{\eq\@equation*}%
6712 \eq\markline@amsthm@register{equation*}%
6713 \eq\amsmath@before{\eq\provide@undefineenv{equation*}}%
6714 \ifdefined\eq\tagging@on
6715 \AddToHook{begindocument/end}{%
6716 \eq\provide@moveenv{equation*}{\eq\@equation*}%
6717 \eq\markline@amsthm@register{equation*}%
6718 }%
6719 \fi
6720 \fi
6721 }

```

`\provide@displaymath` **TODO:** describe

```

6722 \def\eq\provide@displaymath#1{%
6723 \eq\provide@onlyonce{displaymath}{#1}%
6724 \ifdefined\eq\@tmp
6725 \eq\provide@moveenv{\eq\@tmp}{\eq\@displaymath}%
6726 \eq\markline@amsthm@register{\eq\@tmp}%
6727 \eq\tagging@register@luamml{\eq\@tmp}%
6728 \else
6729 \eq\provide@moveenv{displaymath}{\eq\@displaymath}%
6730 \eq\markline@amsthm@register{displaymath}%
6731 \ifdefined\eq\tagging@on
6732 \AddToHook{begindocument/end}{%
6733 \eq\provide@moveenv{displaymath}{\eq\@displaymath}}%
6734 \fi
6735 \fi
6736 }

```

`\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:

```

6737 \def\eq\provide@subequations#1{%
6738 \eq\provide@onlyonce{subequations}{#1}%
6739 \ifdefined\eq\@tmp
6740 \eq\provide@moveenv{\eq\@tmp}{\eq\@subequations}%
6741 \else
6742 \eq\amsmath@after{%
6743 \eq\provide@moveenv{subequations}{\eq\@subequations}%
6744 }%
6745 \AddToHook{package/hyperref/after}{%
6746 \AddToHook{cmd/subequations/before}[hyperref]{}%
6747 \AddToHook{cmd/subequations/after}[hyperref]{}%
6748 \RemoveFromHook{cmd/subequations/before}[hyperref]%
6749 \RemoveFromHook{cmd/subequations/after}[hyperref]%
6750 \AddToHook{begindocument/end}{%
6751 \eq\provide@moveenv{subequations}{\eq\@subequations}}%

```

```

6752 }%
6753 \eql@provide@moveenv{subequations}{\eql@subequations}%
6754 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6755 \fi
6756 }

```

`\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:

```

6757 \def\eql@provide@sqr{%
6758   \let\[\eql@sqr@open
6759   \let\]\eql@sqr@close
6760   \eql@amsmath@after{%
6761     \let\[\eql@sqr@open
6762     \let\]\eql@sqr@close
6763   }%
6764   \ifdefined\eql@tagging@on
6765     \AddToHook{begindocument/end}{%
6766       \let\[\eql@sqr@open
6767       \let\]\eql@sqr@close
6768     }%
6769   \fi
6770 }

```

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

```

6771 \def\eql@provide@ang{%
6772   \let\<\eql@ang@open
6773   \let\>\eql@ang@close
6774 }

```

TODO: describe

```

6775 \def\eql@provide@tagform{%
6776   \def\tagform##1{\maketag@@@{\eql@tags@tagform{##1}}}
6777 \def\eql@provide@maketag{%
6778   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}

```

Interface.

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

```

6779 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
6780 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
6781 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
6782 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}
6783 \eql@define@key{provide}{multline}[]{\eql@provide@amsenv{multline}{#1}}
6784 \eql@define@key{provide}{align}[]{\eql@provide@amsenv{align}{#1}}
6785 \eql@define@key{provide}{flalign}[]{\eql@provide@amsenv{flalign}{#1}}
6786 \eql@define@key{provide}{alignat}[]{\eql@provide@amsenv{alignat}{#1}}
6787 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsenv{xalignat}{#1}}
6788 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsenv{xxalignat}{#1}}
6789 \eql@define@key{provide}{gather*}[]{\eql@provide@amsenv{gather*}{#1}}
6790 \eql@define@key{provide}{multline*}[]{\eql@provide@amsenv{multline*}{#1}}
6791 \eql@define@key{provide}{align*}[]{\eql@provide@amsenv{align*}{#1}}
6792 \eql@define@key{provide}{flalign*}[]{\eql@provide@amsenv{flalign*}{#1}}

```



```

6793 \eqld@define@key{provide}{alignat*}[]{\eqld@provide@amsenv{alignat*}{#1}}
6794 \eqld@define@key{provide}{xalignat*}[]{\eqld@provide@amsenv{xalignat*}{#1}}
6795 \eqld@define@key{provide}{gathered}[]{\eqld@provide@amsbox{gathered}{#1}}
6796 \eqld@define@key{provide}{multlined}[]{\eqld@provide@multlined{#1}}
6797 \eqld@define@key{provide}{aligned}[]{\eqld@provide@amsbox{aligned}{#1}}
6798 \eqld@define@key{provide}{alignedat}[]{\eqld@provide@amsbox{alignedat}{#1}}
6799 \eqld@define@key{provide}{cases}[]{\eqld@provide@matrix{cases}\eqld@false{#1}}
6800 \eqld@define@key{provide}{matrix}[]{\eqld@provide@matrix{matrix}\eqld@false{#1}}
6801 \eqld@define@key{provide}{pmatrix}[]{\eqld@provide@matrix{pmatrix}\eqld@false{#1}}
6802 \eqld@define@key{provide}{bmatrix}[]{\eqld@provide@matrix{bmatrix}\eqld@true{#1}}
6803 \eqld@define@key{provide}{Bmatrix}[]{\eqld@provide@matrix{Bmatrix}\eqld@true{#1}}
6804 \eqld@define@key{provide}{vmatrix}[]{\eqld@provide@matrix{vmatrix}\eqld@true{#1}}
6805 \eqld@define@key{provide}{Vmatrix}[]{\eqld@provide@matrix{Vmatrix}\eqld@true{#1}}
6806 \eqld@define@key{provide}{subequations}[]{\eqld@provide@subequations{#1}}
6807 \eqld@define@key{provide}{sqr}[]{\eqld@provide@sqr}
6808 \eqld@define@key{provide}{ang}[]{\eqld@provide@ang}
6809 \eqld@define@key{provide}{eqref}[]{\eqld@provide@eqref{#1}}
6810 \eqld@define@key{provide}{tagform}[]{\eqld@provide@tagform}
6811 \eqld@define@key{provide}{maketag}[]{\eqld@provide@maketag}

```

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

```

6812 \newcommand{\eqnlinesprovide}[1]{%
6813   \eqld@setkeys{provide}{#1}%
6814   \ignorespaces
6815 }

```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6816 \let\eqld@error@packageoption\@gobble
```

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

```

6817 \DeclareOption{leqno}{\eqnlinesset{tagsleft}}
6818 \DeclareOption{fleqn}{\eqnlinesset{left}}

```

Pass undeclared options on to keyval processing:

```
6819 \DeclareOption*{\expandafter\eqnlinesset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

6820 \eqld@defaults@eqnlines
6821 \eqld@equations@columns@set
6822 \eqld@box@columns@set

```

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

```

6823 \ifdefined\tagsleft@true\else
6824   \expandafter\newif\csname iftagsleft@\endcsname
6825 \fi
6826 \ifdefined\@fleqntrue\else
6827   \expandafter\newif\csname if@fleqn\endcsname
6828 \fi

```

Import amsmath switches leqno as tagsleft and fleqn as left:

```

6829 \eq@amsmath@after{%
6830   \ifnum\eq@provide@opt@env=\tw@
6831     \iftagsleft@
6832       \eqnlineset{tags=left}%
6833     \else
6834       \eqnlineset{tags=right}%
6835     \fi
6836   \if@fleqn
6837     \eqnlineset{layout=left}%
6838   \else
6839     \eqnlineset{layout=center}%
6840   \fi
6841 \fi
6842 }

```

Process package options:

```

6843 \ProcessOptions

```

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

```

6844 \def\eq@error@packageoption#1{%
6845   \eq@error{may only use '#1' as a package option}%
6846 }

```

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

```

6847 \ifnum\eq@provide@opt@env=\tw@
6848 \ifdefined\eq@provide@opt@matrix
6849   \let\eq@provide@opt@amsmathpatch\eq@false
6850 \fi\fi
6851 \ifdefined\eq@provide@opt@backup
6852   \let\eq@provide@opt@amsmathpatch\eq@true
6853 \fi
6854 \ifdefined\eq@provide@opt@amsmathpatch
6855   \eq@amsmath@fixends
6856   \eq@amsmath@fixmatrices
6857 \fi

```

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

```

6858 \ifdefined\eq@provide@opt@backup\eq@provide@backup\fi

```

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

```

6859 \ifnum\eq@provide@opt@env>\z@
6860   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6861 \fi

```

Provide amsmath equation environments if desired:

```

6862 \ifnum\eq@provide@opt@env=\tw@
6863   \eqnlinesprovide{%
6864     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6865     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6866     multlined,gathered,aligned,alignedat,%
6867     subequations}
6868 \fi

```

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

```
6869 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
6870 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```

Provide `cases` and `matrix` environments if desired:

```
6871 \ifdefined\eql@provide@opt@matrix
```

```
6872   \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
```

```
6873 \fi
```