

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`

v1.0.1: 2026-03-23

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

Abstract

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

Contents

1	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	127
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\eql@verbose@on{%
2   \def\eql@verbose@info##1{\PackageInfo{eqnlines}{##1}}%
3   \def\eql@verbose@infoarg##1##2{\eql@verbose@info{##1##2}}%
4 }
5 \def\eql@verbose@off{%
6   \let\eql@verbose@info\@gobble
7   \let\eql@verbose@infoarg\@gobbletwo
8 }
9 \eql@verbose@off
```

TODO: describe

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

2.2 Supporting Definitions

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

```
19 \let\eql@false\@undefined
20 \let\eql@true\@empty
```

TODO: for comparison within `\ifx`

```
21 \def\eql@relax{\relax}
```

TODO: describe

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

2.3 Dollardollar Abstraction

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

```
26 \ifdefined\dollar@dollar@begin
27   \def\eql@dollar@dollar@begin{\dollar@dollar@begin}
28   \def\eql@dollar@dollar@end{\dollar@dollar@end}
29 \else
30   \def\eql@dollar@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\eq@speggroup{\ifnum0='{\fi\iffalse}\fi}
100 \def\eq@srbgroup{\relax\iffalse{\fi\ifnum0='{}\fi}
101 \def\eq@sregroup{\let\@let@token\relax\ifnum0='{\fi\iffalse}\fi}

```

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

```

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

```

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

```

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

```

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

```

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

```

2.5 Error Messages

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

```

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

```

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

```

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

```

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

```

\eq@warn@label@multiple
\eq@warn@tag@multiple
\eq@warn@name@multiple
\eq@warn@ref@multiple

```

```

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

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

```

2.6 amsmath Integration

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

```

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

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

```

165 \let\eq@tagging@on\eq@false
166 \IfFormatAtLeastTF{2025-06-01}{%
167   \csname tag_if_active:T\endcsname{\let\eq@tagging@on\eq@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@tablesaveinner{%
183     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
184   \def\eql@tagging@tableaddinner{%
185     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
186   \def\eql@tagging@tablesavealign{%
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:Nf\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     \eql@letcs{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@tablesavealign\@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 `\eql@scan@body` the tokens so that special commands such as `\verb` can be processed properly. The register `\eql@scan@body` holds the currently selected mode of operation: **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\csname\@currenvir\endcsname}%
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 }

```

`\an@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
```

`\eql@display@linewidth` **TODO:** describe
`\eql@display@marginleft`
`\eql@display@marginright` 458 `\let\eql@display@linewidth\@undefined`
459 `\let\eql@display@marginleft\@undefined`
460 `\let\eql@display@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` 465 `\def\eql@break@line@sep{2em minus 1em}`
`\eql@break@col@shortsep` 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`

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

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

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`

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

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

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

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

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

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

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

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

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

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

```
508 \newdimen\eql@tagwidthmin@
509 \eql@tagwidthmin@\z@
```

`\eql@tagsepmin@ (dimen)` The minimum separation between an equation and its tag is given by `\eql@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\eql@tagsepmin@
511 \def\eql@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*)

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

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

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

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

```
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*)

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

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

```
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*)

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

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

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

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

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

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

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

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

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

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

```

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@display@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 `\eql@punct@line` value indicates that the punctuation should be handed down to the next lower level:

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

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

`\eql@punct@cases`

```

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

```

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

```

590 \let\eq@punct@class\@empty
591 \let\eq@punct@sep\@empty

```

```

\eq@punct@top@set TODO: describe
\eq@punct@top@stop
\eq@punct@top@reset
592 \def\eq@punct@top@set{\let\eq@punct@top\eq@punct@block}
593 \def\eq@punct@top@stop{\let\eq@punct@top\relax}
594 \def\eq@punct@top@reset{\let\eq@punct@top\@undefined}

```

`\eq@punct@set` **TODO:** describe

```

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

```

Set the puncton 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\eq@punct@adopt{%
612   \eq@ifstar@tight\eq@punct@adopt@relax\eq@punct@adopt@set}
613 \def\eq@punct@adopt@set#1{\eqnadopt{punct={#1}}\ignorespaces}
614 \def\eq@punct@adopt@relax{\eqnadopt{punct*}\ignorespaces}

```

TODO: describe

```

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

```

`\eqnpunct` **TODO:** describe

```

622 \let\eqnpunct\eq@punct@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     \ifmode\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:

`\eql@punct@print@line`

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

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

`\eql@punct@apply@line`

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

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

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

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

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

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

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

4.2 Math Classes at Alignment

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

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

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

`\eql@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\eql@class@innerleft@set#1{%
687   \def\eql@class@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@rel symb
841   \fi
842   \ifx\eql@parseopt@token;%
843     \let\eql@parseopt@next\eql@parseopt@rel cont
844     \fi
845     \ifx\eql@parseopt@token:%
846       \let\eql@parseopt@next\eql@parseopt@rel start
847       \fi
848       \ifx\eql@parseopt@token|%
849         \let\eql@parseopt@next\eql@parseopt@rel ord
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.eql-\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\eq@numbering@tab@t\eq@numbering@tab@top
961 \let\eq@numbering@tab@b\eq@numbering@tab@bottom
962 \let\eq@numbering@tab@c\eq@numbering@tab@center
963 \let\eq@numbering@tab@m\eq@numbering@tab@median
964 \eq@letcs{eq@numbering@tab@+}\eq@numbering@tab@best
965 \eq@letcs{eq@numbering@tab@m*}\eq@numbering@tab@middle
966 \eq@letcs{eq@numbering@tab@c*}\eq@numbering@tab@baseline
967 \eq@letcs{eq@numbering@tab@c!}\eq@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

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

```

TODO: describe

```

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

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

1034 \let\eq@numbering@init@median\eq@numbering@init@continuous
1035 \def\eq@tagpos@eval@median{%
1036 \ifnum\eq@tagpos@row@=z@
1037 \ifodd\eq@totalrows@
1038 \eq@tagpos@row@\numexpr(\eq@totalrows@+\@ne)/\tw@\relax
1039 \else
1040 \eq@tagpos@row@\numexpr(\eq@totalrows@+\tw@)/\tw@\relax
1041 \eq@dimensions@get\eq@tagpos@row@
1042 \advance\eq@tagpos@shift@\dimexpr\eq@line@height@
1043 +(\eq@line@interline@-\eq@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@amprotect\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@printsubeqlabel
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\eq@numbering@print@line@eval{%
1378   \ifdefined\eq@numbering@multi
1379     \if@eqnsw
1380       \eq@tags@container
1381     \fi
1382   \else
1383     \ifnum\eq@tagpos@row@=\eq@row@
1384       \@eqnswtrue
1385       \eq@tags@container@block
1386     \else
1387       \@eqnswfalse
1388     \fi
1389   \fi
1390 }
```

5.5 Positioning

TODO: describe

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

TODO: describe

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

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

TODO: describe

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

TODO: describe

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

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

TODO: describe

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

TODO: describe

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

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

```

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

```

ql@tagbox@print@cell

```

1819 \def\eq\tagbox@print@cell{%
1820 \eq\tagging@tagaddbox
1821 \ifdefined\eq\tagleft
1822 \ifdefined\eq\tagpos@reserve
1823 \ifdim\eq\tagwidth@>\dimexpr\eq\line@avail@+\eq\tagfuzz@\relax
1824 \let\eq\tagpos@reserve\eq\false
1825 \fi
1826 \fi
1827 \if@eqnsw
1828 \eq\tagbox@print@tagleft
1829 \fi
1830 \kern\displaywidth
1831 \else
1832 \kern\displaywidth
1833 \ifdefined\eq\tagpos@reserve
1834 \ifdim\eq\tagwidth@>%
1835 \dimexpr\displaywidth-\eq\line@width@+\eq\tagfuzz@\relax
1836 \let\eq\tagpos@reserve\eq\false
1837 \fi
1838 \fi
1839 \if@eqnsw
1840 \eq\tagbox@print@tagright
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 \eq\amsmath@undefine\c@parentequation
1845 \eq\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\eq\subequations@template{\theparentequation\alph{equation}}
1850 \def\eq\subequations@template@set#1{\def\eq\tmp##1##2{#1}%
1851 \edef\eq\subequations@template{%
1852 \unexpanded\expandafter{\eq\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
```

`ql@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 }
```

`l@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

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

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

```
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
```

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

```
1975 \def\eql@getdsp@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\eql@displaybreak@default{%
1979   \eql@warning{Invalid use of \string\displaybreak}{}%
1980   \eql@teststaropt@loose@gobble\eql@gobbleopt{}}
1981 \eql@amsmath@after{\let\eql@displaybreak@default\displaybreak}
1982 \eql@amsmath@let\displaybreak\eql@displaybreak@default
```

```
1983 \newcount\eql@displaybreak@pen@
1984 \newcount\eql@displaybreak@prepen@
1985 \newcount\eql@displaybreak@postpen@
```

TODO: describe

```
1986 \protected\def\eql@displaybreak{%
1987   \relax
1988   \eql@ampprotecttwo\eql@teststaropt@tight
1989   \eql@displaybreak@star\eql@displaybreak@level{4}%
1990 }
```

```
1991 \def\eql@displaybreak@star#1{%
1992   \global\eql@appendexpand\eql@interline@container{%
1993     \eql@displaybreak@pen@\the\numexpr#1\relax\relax}%
1994 }
```

```
1995 \def\eql@displaybreak@level[#1]{%
1996   \ifnum#1<\z@
1997     \global\eql@append\eql@interline@container{\eql@displaybreak@pen@\@MM}%
1998   \else
1999     \global\eql@appendexpand\eql@interline@container{%
2000       \eql@displaybreak@pen@-\@getpen{#1}\relax}%
2001   \fi
2002 }
```

TODO: describe

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

TODO: describe

```
2010 \def\eql@displaybreak@post#1{%
```

```

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

```

TODO: describe

```

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

```

7.2 Explicit Vertical Space

TODO: describe

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

```

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

```

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

```

2033 \def\eql@vspace@line{%
2034   \eql@ifstar@loose\eql@vspace@addfixedbefore\eql@vspace@add
2035 }
2036 \def\eql@vspace@add#1{%
2037   \global\eql@appendexpand\eql@interline@container{%
2038     \advance\eql@vspaceskip@the\glueexpr#1\relax\relax}}
2039 \def\eql@vspace@addfixedbefore#1{%
2040   \global\eql@appendexpand\eql@interline@container{%
2041     \noexpand\eql@append\noexpand\eql@display@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\eql@vspace@addfixedafter#1{%
2050   \global\eql@appendexpand\eql@interline@container{%
2051     \noexpand\eql@append\noexpand\eql@display@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     \@tempdimb\dimexpr
2106       \eqL@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
2107     \@tempdima\dimexpr
2108       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
2109     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
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\normallineskip
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@apply\@empty
2129 \let\eqL@beamerbasecolor@avoid\@empty
2130 \AddToHook{package/beamerbasecolor/after}{%
2131   \def\eqL@beamerbasecolor@apply{%
2132     \donotcolorouterdisplaymaths
2133     \donotcoloroutermaths
2134     \beamer@setdisplaymathcolor
2135   }%
2136   \def\eqL@beamerbasecolor@avoid{\donotcoloroutermaths}%
2137 }

```

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

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

```

2138 \newskip\eqL@abovespace@
2139 \newskip\eqL@belowspace@

```

`\eqL@display@enter`

```

2140 \def\eqL@display@enter{%
2141   \if@noskipsec\leavevmode\par\fi
2142   \ifvmode
2143     \eqL@prevdepth@\prevdepth
2144     \nointerlineskip
2145     \noindent

```

```

2146 \else
2147   \eql@prevdepth@\maxdimen
2148 \fi
2149 \eql@beamerbasecolor@apply
2150 }

```

`\eql@display@adjust`

```

2151 \def\eql@display@adjust{%
2152   \ifdefined\eql@display@linewidth
2153     \displaywidth\glueexpr\eql@display@linewidth\relax
2154     \advance\displaywidth-\displayindent
2155   \fi
2156   \ifdefined\eql@display@marginleft
2157     \advance\displaywidth\displayindent
2158     \displayindent\glueexpr\eql@display@marginleft\relax
2159     \advance\displaywidth-\displayindent
2160   \fi
2161   \ifdefined\eql@display@marginright
2162     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
2163   \fi
2164   \ifdim\displaywidth<\z@
2165     \displaywidth\z@
2166   \fi
2167 }

```

`\eql@display@init`

```

2168 \def\eql@display@init{%
2169   \let\eql@display@restore\eql@display@restore@active
2170   \let\displaybreak\eql@displaybreak
2171   \let\eql@vspace@org\vspace
2172   \let\vspace\eql@vspace
2173   \let\eqncontrol\eql@control
2174   \let\eql@display@injectbefore\@empty
2175   \let\eql@display@injectafter\@empty
2176   \let\eqnpunct\eql@punct@setnext
2177   \eql@spread@set
2178   \eql@strut@make
2179   \let\eql@frame@cmd\@undefined
2180 }

```

`\eql@display@print`

```

2181 \def\eql@display@print{%
2182   \eql@punct@top@set
2183   \let\eql@display@container\@empty
2184   \eql@display@firstavail@\z@
2185   \eql@display@aboveextend@\z@
2186   \eql@display@belowextend@\z@
2187   \global\let\eql@interline@container\eql@interline@container@clear
2188 }

```

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

```

2189 \def\eql@display@halign@init#1{%
2190   \eql@row@\z@
2191   \eql@prevgraf@\prevgraf
2192   \everycr{\noalign{%

```

```

2193 \global\advance\eq@row@\@ne
2194 \prevgraf\numexpr\prevgraf+\@ne\relax
2195 #1%
2196 }}%
2197 }

```

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

```

2198 \def\eq@display@halign@start{%
2199 \prevgraf\numexpr\eq@prevgraf+\@ne\relax
2200 \ifdim\eq@prevdepth@=\maxdimen\else
2201 \prevdepth\eq@prevdepth@
2202 \fi
2203 \ifdim\prevdepth=-\@m\p@\else
2204 \ifdefined\eq@display@height
2205 \skip@\baselineskip
2206 \advance\skip@-\glueexpr\eq@display@height\relax
2207 \advance\skip@-\prevdepth\relax
2208 \ifdim\skip@<\lineskiplimit
2209 \skip@\lineskip
2210 \fi
2211 \advance\skip@-\eq@spread@\relax
2212 \vskip\skip@
2213 \nointerlineskip
2214 \else
2215 \vskip-\eq@spread@\relax
2216 \fi
2217 \fi
2218 }

```

TODO: describe

```

2219 \def\eq@display@vspace{%
2220 \advance\abovedisplayskip\eq@abovespace@
2221 \advance\belowdisplayskip\eq@belowspace@
2222 }

```

TODO: describe

```

2223 \def\eq@display@vspace@native{%
2224 \advance\abovedisplayskip\eq@abovespace@
2225 \advance\belowdisplayskip\eq@belowspace@
2226 \advance\abovedisplayshortskip\eq@abovespace@
2227 \advance\belowdisplayshortskip\eq@belowspace@
2228 }

```

TODO: describe

```

2229 \def\eq@display@penalty{%
2230 \ifnum\eq@displaybreak@postpen@=\@MM\else
2231 \postdisplaypenalty\eq@displaybreak@postpen@
2232 \fi
2233 \ifnum\eq@displaybreak@pen@=\@MM\else
2234 \postdisplaypenalty\eq@displaybreak@pen@
2235 \fi
2236 \ifnum\eq@displaybreak@prepen@=\@MM\else
2237 \predisdisplaypenalty\eq@displaybreak@prepen@
2238 \fi
2239 }

```

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

```

2240 \def\eqldisplay@halign@end{%
2241   \eql@interline@container
2242   \eqldisplay@injectbefore
2243   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
2244   \ifdefined\eqldisplay@depth
2245     \prevdepth\glueexpr\eqldisplay@depth\relax
2246   \fi
2247 }

```

\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?

```

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

```

```

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

```

```

2352 \fi
2353 \advance\abovedisplayskip-\eql@display@aboveextend@relax
2354 \ifdim\abovedisplayskip<\skip@
2355   \abovedisplayskip\skip@
2356 \fi
2357 \skip@\glueexpr\eql@skip@tag@belowrelax
2358 \ifdim\skip@>\belowdisplayskip
2359   \skip@\belowdisplayskip
2360 \fi
2361 \ifdim\eql@display@belowextend@>\z@
2362   \advance\belowdisplayskip-\eql@display@belowextend@relax
2363   \ifdim\belowdisplayskip<\skip@
2364     \belowdisplayskip\skip@
2365   \fi
2366 \fi
2367 }

```

TODO: describe

```

2368 \def\eql@display@leave{%
2369   \prevgraf\eql@prevgraf@
2370   \ifcase\eql@skip@mode@leave@
2371     \or
2372     \endgraf
2373     \or
2374     \endgraf
2375     \prevdepth-\@m\p@
2376 \fi
2377 }

```

TODO: describe

```

2378 \def\eql@display@nest{%
2379   \let\displaybreak\eql@displaybreak@default
2380   \let\intertext\eql@intertext@default
2381   \let\vspace\eql@vspace@org
2382 }

```

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

```

2383 \def\eql@display@restore@active{%
2384   \let\label\eql@label@org
2385   \let>tag\eql>tag@default
2386   \let\raisetag\eql\raisetag@default
2387   \let\displaybreak\eql@displaybreak@default
2388   \let\intertext\eql@intertext@default
2389   \let\vspace\eql@vspace@org
2390   \ifdefined\eql@amp@mode
2391     \let&\eql@amp@org
2392 \fi
2393 \let\eqnpunct\eql@punct@adopt
2394 \let\eql@punct@block\@undefined
2395 \let\eql@display@restore\@empty
2396 }

```

TODO: describe

```

2397 \let\eql@display@restore\@empty
2398 \eql@append\@arrayparboxrestore{%
2399   \eql@display@restore
2400   \ifdefined\eql@ampproof@active

```

```

2401 \eql@amprevert
2402 \fi
2403 \@displayfalse
2404 }

```

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)

```

2405 \def\eql@stack@enable{%
2406 \let\eql@stack@save@equations\eql@stack@save@equations@
2407 \let\eql@stack@save@box\eql@stack@save@box@
2408 }

```

TODO: describe

```

2409 \let\eql@stack@save@equations\eql@stack@enable
2410 \let\eql@stack@save@box\eql@stack@enable
2411 \let\eql@stack@restore\@empty

```

TODO: describe

```

2412 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
2413 \def\eql@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

TODO: further global variables: global registers: `\eql@nextopt`, `\eql@tags@glabel@` used locally without possibility of change between setting and retrieving:
`\eql@prevgraf@`, `\eql@skip@mode@leave@`, `\eql@shape@lastrow`, `\eql@frame@prevcmd`
TODO: to be reviewed: `\eql@intertext@after`, `\eql@intertext@opt` **TODO:** describe

```

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

```

```

2441 \eql@stack@save@reg\eql@tagwidth@max@
2442 \eql@stack@save@reg\eql@tagpos@row@
2443 \eql@stack@save@reg\eql@row@
2444 \eql@stack@save@reg\eql@tagrows@
2445 }%
2446 }

```

TODO: describe

```

2447 \def\eql@stack@save@box@{%
2448 \let\eql@stack@cell@container\eql@cell@container
2449 \edef\eql@stack@restore@{%
2450 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2451 \eql@stack@save@reg\eql@row@
2452 }%
2453 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2454 \def\eql@measure@init#1#2{%
2455 \eql@dimensions@reset
2456 \let\eql@display@container\@empty
2457 \eql@numbering@measure@init
2458 \eql@row@z@
2459 \eql@totalheight@z@
2460 \eql@totalrows@M
2461 \eql@line@prevdepth@-@m@p@
2462 \eql@line@interline@z@
2463 \tabskipz@skip
2464 \everycr{\noalign{%
2465 \global\advance\eql@row@\@ne
2466 #1%
2467 }}%
2468 \eql@punct@top@set
2469 \global\let\eql@interline@container\eql@interline@container@clear
2470 \eql@measure@savestate
2471 \eql@multi@cr@let{#2}%
2472 }

```

TODO: describe

```

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

```

```
2483 \fi
2484 }
```

TODO: describe

```
2485 \def\eq@measure@endrow{%
2486 \ifdim\eq@line@prevdepth@=-\@m\p@\else
2487 \dimen@\dimexpr\baselineskip-\eq@line@height@-\eq@line@prevdepth@\relax
2488 \ifdim\dimen@<\lineskiplimit
2489 \dimen@\lineskip
2490 \fi
2491 \advance\eq@line@interline@\dimen@
2492 \fi
2493 \eq@dimensions@endrow
2494 \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2495 \global\eq@tagwidth@max@\eq@tagwidth@
2496 \fi
2497 \ifdim\eq@tagwidth@>\z@
2498 \global\advance\eq@tagrows@\@ne
2499 \fi
2500 \global\advance\eq@totalheight@\dimexpr
2501 \eq@line@interline@+\eq@line@height@+\eq@line@depth@
2502 \global\eq@line@interline@\z@
2503 \global\eq@line@prevdepth@\eq@line@depth@
2504 }
```

TODO: describe

```
2505 \def\eq@measure@close{%
2506 \advance\eq@row@-\tw@
2507 \eq@totalrows@\eq@row@
2508 \ifnum\eq@totalrows@>\z@
2509 \eq@dimensions@get\@ne
2510 \eq@topheight@\dimexpr\eq@line@height@+\eq@line@interline@\relax
2511 \eq@dimensions@get\eq@totalrows@
2512 \eq@bottomdepth@\eq@line@depth@
2513 \fi
2514 \eq@numbering@measure@blocktag
2515 \begingroup
2516 \eq@tags@container
2517 \if@eqnsw
2518 \eq@tagbox@make\eq@composetag@measure
2519 \ifdefined\eq@tagpos@reserve\else
2520 \eq@tagwidth@\z@
2521 \fi
2522 \eq@dimensions@saveblocktag
2523 \else
2524 \eq@dimensions@savenoblocktag
2525 \eq@numbering@warnunused
2526 \fi
2527 \endgroup
2528 \eq@dimensions@get\z@
2529 \eq@measure@restorestate
2530 }
```

measure@restorestate

eq@measure@savestate

```
2531 \let\eq@measure@restorestate\@empty
2532 \def\eq@measure@savestate{%
2533 \begingroup
```

```

2534 \def\@elt##1{%
2535 \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2536 \global\edef\@gtempa{\cl@ckpt}%
2537 \endgroup
2538 \let\eq@measure@restorestate\@gtempa
2539 }

```

8.2 Line Breaks

TODO: describe

`\eq@multi@cr`

```

2540 \def\eq@multi@cr{%
2541 \let\eq@punct@term\eq@false
2542 \let\eq@class@rel@composed\@empty
2543 \eq@ampprotect\eq@multi@cr@test\eq@multi@cr@process}

```

TODO: describe

```

2544 \def\eq@multi@cr@test@setopt{%
2545 \let\eq@multi@cr@test\eq@multi@cr@testopt}
2546 \def\eq@multi@cr@test@setall{%
2547 \let\eq@multi@cr@test\eq@multi@cr@testall}

```

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

```

2548 \def\eq@multi@cr@testopt#1{\eq@teststaropt@tight
2549 {\eq@displaybreak@star\@M\eq@multi@cr@testopt@set{#1}}%
2550 {\eq@multi@cr@testopt@set{#1}}{Opt}}
2551 \def\eq@multi@cr@testopt@set#1[#2]{\eq@vspace@add{#2}#1}

```

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

`\eq@multi@cr@testall@parse`

```

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

```

```

2576 \let\eql@parseopt@next\eql@multi@cr@parse@break
2577 \fi
2578 \ifx\eql@parseopt@token=%
2579 \let\eql@parseopt@next\eql@parseopt@relsymp
2580 \fi
2581 \ifx\eql@parseopt@token;%
2582 \let\eql@parseopt@next\eql@parseopt@relcont
2583 \fi
2584 \ifx\eql@parseopt@token:~
2585 \let\eql@parseopt@next\eql@parseopt@relstart
2586 \fi
2587 \ifx\eql@parseopt@token|~
2588 \let\eql@parseopt@next\eql@parseopt@relord
2589 \fi
2590 \ifx\eql@parseopt@token?~
2591 \let\eql@parseopt@next\eql@multi@cr@parse@rel
2592 \fi
2593 \ifx\eql@parseopt@token&~
2594 \let\eql@parseopt@next\eql@parseopt@end
2595 \fi
2596 }
2597 \def\eql@multi@cr@parse@vspace[#1]{\eql@vspace@add{#1}\eql@parseopt@peek}
2598 \def\eql@multi@cr@parse@star#1{\eql@displaybreak@star\M\eql@parseopt@peek}
2599 \def\eql@multi@cr@parse@break{\numbernext\eql@parseopt@punctclear}
2600 \def\eql@multi@cr@parse@rel#1#2{%
2601 \def\eql@tmp{#2}%
2602 \ifx\eql@tmp\eql@relax\else
2603 \ifdefined\eql@punct@next\else
2604 \eql@punct@next@clear
2605 \fi
2606 \ifdefined\eql@multi@cr@relnext\numbernext\fi
2607 \fi
2608 \ifdefined\eql@multi@linesmode
2609 \ifx\eql@tmp\@empty
2610 \def\eql@class@rel@composed{\eql@shape@cont}%
2611 \else
2612 \def\eql@class@rel@composed{\eql@shape@rel#2}%
2613 \fi
2614 \else
2615 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
2616 \fi
2617 \eql@parseopt@end}

```

eql@multi@cr@process

```

2618 \def\eql@multi@cr@process{%
2619 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
2620 \edef\eql@tmp{%
2621 \unexpanded{%
2622 \eql@multi@endline
2623 \cr
2624 \eql@multi@cr@interline
2625 }%
2626 \unexpanded\expandafter{\eql@class@rel@composed}%
2627 }%
2628 \eql@tmp
2629 }

2630 \def\eql@multi@cr@interline{%

```

```

2631 \noalign{%
2632   \eql@interline@container
2633   \eql@display@injectbefore
2634   \ifnum\eql@displaybreak@pen@=\@MM
2635     \penalty\interdisplaylinepenalty
2636   \else
2637     \penalty\eql@displaybreak@pen@
2638   \fi
2639   \vskip\eql@vspaceskip@
2640   \global\advance\eql@line@interline@\eql@vspaceskip@
2641   \eql@display@injectafter
2642   \global\let\eql@interline@container\eql@interline@container@clear
2643 }%
2644 }

```

`\eql@multi@cr@let`

```

2645 \def\eql@multi@cr@let#1{%
2646   \let\\eql@multi@cr
2647   \let\eql@multi@endline#1%
2648 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2649 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2650 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2651 \def\eql@intertext@process{%
2652   \eql@multi@endline
2653   \cr
2654   \ifmeasuring@
2655     \expandafter\@gobble
2656   \else
2657     \expandafter\eql@intertext@print
2658   \fi
2659 }

```

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

```

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

```

```

2674 \or
2675   \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2676 \or
2677   \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2678 \or
2679   \advance\eql@vspaceskip@\z@skip
2680 \or
2681   \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2682 \or
2683   \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2684 \fi
2685 \vskip\eql@vspaceskip@
2686 \global\let\eql@interline@container\eql@interline@container@clear
2687 \vbox{%
2688   \@parboxrestore
2689   \ifdim
2690     \ifdim@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2691   \else
2692     \parshape\@ne
2693     \@totalleftmargin\linewidth
2694   \fi
2695   \noindent
2696   \prevgraf\eql@prevgraf@
2697   \ignorespaces
2698   #1%
2699   \par
2700   \global\eql@prevgraf@\prevgraf
2701 }%
2702 \penalty\predisplaypenalty
2703 \ifcase\eql@skip@force@above\relax
2704   \vskip\glueexpr\eql@skip@long@above\relax
2705 \or
2706   \vskip\glueexpr\eql@skip@short@above\relax
2707 \or
2708   \vskip\glueexpr\eql@skip@cont@above\relax
2709 \or
2710   \vskip\glueexpr\eql@skip@par@above\relax
2711 \or
2712   \vskip\glueexpr\eql@skip@top@above\relax
2713 \or
2714   \vskip\z@skip
2715 \or
2716   \vskip\glueexpr\eql@skip@med@above\relax
2717 \or
2718   \vskip\glueexpr\eql@skip@custom@above\relax
2719 \fi
2720 %   \eql@prevdepth@\maxdimen
2721 \eql@prevdepth@\z@
2722 \eql@display@halign@start
2723 }
2724 }

```

TODO: describe

```

2725 \newenvironment{eql@intertext}{%
2726   \eql@testopt@tight\eql@intertext@{}}%
2727 }{%
2728   \aftergroup\eql@intertext@after
2729   \ignorespacesafterend

```

2730 }

TODO: describe

```
2731 \def\eql@intertext@env{intertext}
2732 \def\eql@intertext@[#1]{%
2733   \global\def\eql@intertext@opt{#1}%
2734   \ifx\@currenvir\eql@intertext@env
2735     \def\eql@scan@call{\eql@intertext@inject\eql@scan@end}%
2736     \expandafter\eql@scan@env
2737   \else
2738     \expandafter\eql@intertext@process
2739   \fi
2740 }
```

TODO: describe

```
2741 \def\eql@intertext@inject{%
2742   \global\edef\eql@intertext@after{%
2743     \noexpand\eql@intertext@process{%
2744       \ifx\eql@scan@body\eql@scan@body@dump
2745         \eql@scan@body@dump
2746       \else
2747         \noexpand\scantokens{\eql@scan@body@dump}%
2748       \fi
2749     }%
2750   }%
2751 }
```

8.4 Line Marks

TODO: describe

```
2752 \def\eql@markline@pos@below{below}
2753 \def\eql@markline@pos@bottom{bottom}
2754 \def\eql@markline@pos@baseline{baseline}
2755 \let\eql@markline@pos\eql@markline@pos@baseline
2756 \let\eql@markline@shift\z@
2757 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2758 \def\eql@markline@symbol{}
```

TODO: describe

```
2759 \def\eql@markline@select#1{%
2760   \let\eql@markline@shift\z@
2761   \eql@setkeys{markline}{#1}%
2762   \eql@markline@print
2763 }
```

TODO: describe

```
2764 \def\eql@markline@print{%
2765   \dimen@=\dimexpr\eql@markline@shift\relax
2766   \ifx\eql@markline@pos\eql@markline@pos@below
2767     \ifdim\dimen@=\z@\else
2768       \penalty\@M
2769       \vskip-\dimen@
2770     \fi
2771     \nointerlineskip
2772     \penalty\@M
```

```

2773   \vbox{\hfill\hbox{\eql@markline@symbol}}%
2774 \else
2775   \ifx\eql@markline@pos\eql@markline@pos@baseline
2776     \advance\dimen@\prevdepth
2777   \fi
2778   \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}%
2779   \dimen@\prevdepth
2780   \ht\z@\z@
2781   \dp\z@\z@
2782   \nointerlineskip
2783   \penalty\@M
2784   \vbox{\hfill\box\z@}%
2785   \prevdepth\dimen@
2786 \fi
2787 }

```

TODO: describe

```

2788 \def\eql@markline@inject#1{%
2789   \let\eql@markline@push\eql@false
2790   \ifx\eql@markline@pos\eql@markline@pos@below\else
2791     \ifdefined\eql@tagsleft\else
2792       \ifx\eql@equations@main\eql@multi@main
2793         \ifdefined\eql@numbering@multi
2794           \if@eqnsw
2795             \let\eql@markline@push\eql@true
2796           \fi
2797         \else
2798           \ifnum\eql@row@=\eql@tagpos@row@
2799             \let\eql@markline@push\eql@true
2800           \fi
2801         \fi
2802       \else
2803         \if@eqnsw
2804           \let\eql@markline@push\eql@true
2805         \fi
2806       \fi
2807     \fi
2808   \fi
2809   \ifdefined\eql@markline@push
2810     \global\eql@append\eql@interline@container{%
2811       \eql@append\eql@display@injectbefore{\eql@markline@select{push,#1}}}%
2812   \else
2813     \global\eql@append\eql@interline@container{%
2814       \eql@append\eql@display@injectbefore{\eql@markline@select{#1}}}%
2815   \fi
2816 }

```

TODO: describe

```

2817 \def\eql@markline@amsthm@opt[#1]{\eql@markline@inject{qed,#1}}
2818 \def\eql@markline@amsthm@staropt[#1]{\eql@markline@inject{qed,push,#1}}
2819 \def\eql@markline@amsthm@qed{\eql@teststaropt@tight
2820   \eql@markline@amsthm@staropt\eql@markline@amsthm@opt{}}
2821 \def\eql@markline@amsthm@register#1{\eql@letcs{#1@qed}\eql@markline@amsthm@qed}
2822 \def\eql@markline@amsthm@move#1#2{%
2823   \AddToHook{package/amsthm/after}{%
2824     \eql@letcs{#1@qed}\expandafter}\csname#2@qed\endcsname}}

```

9 Column Placement

TODO: describe

9.1 Supporting Definitions

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

```
2825 \newcount\eql@shape@pos@
2826 \newdimen\eql@shape@amount@
2827 \let\eql@shape@lastrow\eql@false
```

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

```
\eql@marginleft@min@ (dimen)
\eql@marginright@ (dimen)
\centeroffset@ (dimen)
2828 \newdimen\eql@marginleft@
2829 \newdimen\eql@marginright@
2830 \newdimen\eql@marginleft@min@
2831 \newdimen\eql@centeroffset@
```

9.2 Shape Schemes

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

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

```
2832 \def\eql@shape@tab@default{default}
2833 \def\eql@shape@tab@left{left}
2834 \def\eql@shape@tab@center{center}
2835 \def\eql@shape@tab@right{right}
2836 \def\eql@shape@tab@first{first}
2837 \def\eql@shape@tab@hanging{hanging}
2838 \def\eql@shape@tab@steps{steps}
```

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

```
2839 \let\eql@shape@tab@def\eql@shape@tab@default
2840 \let\eql@shape@tab@l\eql@shape@tab@left
2841 \let\eql@shape@tab@cl\eql@shape@tab@center
2842 \let\eql@shape@tab@r\eql@shape@tab@right
2843 \let\eql@shape@tab@rc\eql@shape@tab@first
2844 \let\eql@shape@tab@indent\eql@shape@tab@first
2845 \let\eql@shape@tab@hang\eql@shape@tab@hanging
2846 \let\eql@shape@tab@lcl\eql@shape@tab@hanging
2847 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
2848 \let\eql@shape@tab@lcr\eql@shape@tab@steps
2849 \let\eql@shape@tab@lcr\eql@shape@tab@steps
```

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

```
2850 \let\eql@shape@mode\eql@shape@tab@default
```

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

```
2851 \def\eql@shape@set#1{%
```

```

2852 \ifcsname eql@shape@tab@#1\endcsname
2853   \expandafter\let\expandafter\eql@shape@mode
2854   \csname eql@shape@tab@#1\endcsname
2855 \else
2856   \eql@error{shape '#1' unknown: setting to default}%
2857   \let\eql@shape@mode\eql@shape@tab@default
2858 \fi
2859 }

```

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:

```

2860 \def\eql@shape@layoutcenter@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2861 \def\eql@shape@layoutcenter@center{\eql@shape@pos@ne\eql@shape@amount@z@}
2862 \def\eql@shape@layoutcenter@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
2863 \let\eql@shape@layoutcenter@default\eql@shape@layoutcenter@center
2864 \def\eql@shape@layoutleft@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2865 \def\eql@shape@layoutleft@center{\eql@shape@pos@ne\eql@shape@amount@z@}
2866 \def\eql@shape@layoutleft@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
2867 \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):

```

2868 \def\eql@shape@layoutcenter@first{%
2869   \eql@shape@pos@z@
2870   \eql@shape@amount@z@
2871   \ifnum\eql@totalrows@>\@ne
2872     \ifnum\eql@row@=\@ne
2873       \eql@shape@amount@\eql@indent@
2874     \fi
2875   \fi
2876 }
2877 \def\eql@shape@layoutleft@first{%
2878   \eql@shape@pos@z@
2879   \eql@shape@amount@z@
2880   \ifnum\eql@totalrows@>\@ne
2881     \ifnum\eql@row@=\@ne
2882       \eql@shape@amount@\eql@indent@
2883     \fi
2884   \fi
2885 }

```

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:

```

2886 \def\eql@shape@layoutcenter@hanging{%
2887   \eql@shape@pos@z@
2888   \eql@shape@amount@\eql@indent@
2889   \ifnum\eql@totalrows@>\@ne
2890     \ifnum\eql@row@=\@ne
2891       \eql@shape@amount@z@
2892     \fi
2893   \fi
2894 }
2895 \def\eql@shape@layoutleft@hanging{%
2896   \eql@shape@pos@z@
2897   \eql@shape@amount@z@

```

```

2898 \ifnum\eq@totalrows@>\@ne
2899   \ifnum\eq@row@=\@ne
2900     \eq@shape@amount@-\eq@indent@
2901   \fi
2902 \fi
2903 }

```

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:

```

2904 \def\eq@shape@layoutcenter@steps{%
2905   \eq@shape@amount@z@
2906   \eq@shape@pos@\@ne
2907   \ifnum\eq@totalrows@>\@ne
2908     \ifnum\eq@row@=\@ne
2909       \eq@shape@pos@z@
2910     \fi
2911     \ifnum\eq@row@=\eq@totalrows@
2912       \eq@shape@pos@tw@
2913     \fi
2914   \fi
2915 }
2916 \def\eq@shape@layoutleft@steps{%
2917   \eq@shape@pos@z@
2918   \eq@shape@amount@z@
2919   \ifnum\eq@totalrows@>\@ne
2920     \ifnum\eq@row@=\@ne
2921       \eq@shape@amount@-\eq@indent@
2922     \fi
2923     \ifnum\eq@row@=\eq@totalrows@
2924       \eq@shape@amount@\eq@indent@
2925     \fi
2926   \fi
2927 }

```

`\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`:

```

2928 \let\eq@shape@eval\undefined
2929 \def\eq@shape@select{%
2930   \expandafter\let\expandafter\eq@shape@eval
2931   \csname eq@shape%
2932     @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2933   @\eq@shape@mode\endcsname
2934 }

```

`\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`

```

2935 \protected\def\eq@shape@alignleft{%
2936   \global\eq@append\eq@cell@container{\eq@shape@pos@z@}%
2937   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2938 \protected\def\eq@shape@aligncenter{%
2939   \global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}%
2940   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2941 \protected\def\eq@shape@alignright{%
2942   \global\eq@append\eq@cell@container{\eq@shape@pos@tw@}%
2943   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}

```

```

2944 \def\eq@shape@align@testpar#1{%
2945   \eq@ifstar@tight{#1[\eq@indent]}%
2946   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent]}%
2947   {\eq@testopt@tight{#1}\z@}}
2948 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2949 \protected\def\eq@shape@alignamount{%
2950   \eq@ampprotecttwo\eq@ifstar@tight
2951   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2952 \def\eq@shape@alignamount@add#1{%
2953   \global\eq@appendexpand\eq@cell@container{%
2954     \advance\eq@shape@amount@the\glueexpr#1\relax\relax}}
2955 \def\eq@shape@alignamount@set#1{%
2956   \global\eq@appendexpand\eq@cell@container{%
2957     \eq@shape@amount@the\glueexpr#1\relax\relax}}
2958 \def\eq@shape@align@enable{%
2959   \let\shoveleft\eq@shape@alignleft
2960   \let\shovecenter\eq@shape@aligncenter
2961   \let\shoveright\eq@shape@alignright
2962   \let\shoveby\eq@shape@alignamount
2963 }

```

TODO: describe

```

2964 \protected\def\eq@shape@align@default{%
2965   \eq@warn@here{\shove...}}%
2966   \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2967 \protected\def\eq@shape@alignamount@default{%
2968   \eq@warn@here{\shove...}}%
2969   \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2970 \def\eq@shape@align@disable{%
2971   \let\shoveleft\eq@shape@align@default
2972   \let\shovecenter\eq@shape@align@default
2973   \let\shoveright\eq@shape@align@default
2974   \let\shoveby\eq@shape@alignamount@default
2975 }

```

9.3 Width Data

`\width@block@ (dimen)`

```

2976 \newdimen\eq@tagwidth@block@
2977 \newdimen\eq@tagheight@block@
2978 \newdimen\eq@tagdepth@block@

```

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

```

2979 \let\eq@dimensions@tab\@empty

```

`\eq@dimensions@reset`

```

2980 \def\eq@dimensions@reset{%
2981   \let\eq@dimensions@tab\@empty
2982   \eq@tagwidth@max@\z@
2983   \eq@tagrows@\z@
2984 }

```

\eql@dimensions@add

```
2985 \def\eql@dimensions@add#1{%
2986   \global\eql@appendexpand\eql@dimensions@tab{#1}%
2987 }
```

ql@dimensions@addreg

```
2988 \def\eql@dimensions@addreg#1{#1\the#1\relax}
```

@dimensions@startrow

```
2989 \def\eql@dimensions@startrow{%
2990   \eql@dimensions@add{\eql@dimensions@addreg\eql@row@}%
2991 }
```

@dimensions@savecell

```
2992 \def\eql@dimensions@savecell{%
2993   \eql@dimensions@add{%
2994     \eql@dimensions@addreg\eql@shape@pos@
2995     \eql@dimensions@addreg\eql@cellwidth@
2996     \eql@dimensions@addreg\eql@shape@amount@
2997     \noexpand\eql@dimensions@cellcall
2998   }%
2999 }
```

l@dimensions@savesep

```
3000 \def\eql@dimensions@savesep{%
3001   \eql@dimensions@add{\noexpand\eql@dimensions@sepcall}%
3002 }
```

ql@dimensions@endrow

```
3003 \def\eql@dimensions@endrow{%
3004   \eql@dimensions@add{,%
3005     \eql@dimensions@addreg\eql@tagwidth@
3006     \eql@dimensions@addreg\eql@line@height@
3007     \eql@dimensions@addreg\eql@line@depth@
3008     \eql@dimensions@addreg\eql@line@interline@
3009   };}%
3010 }
```

ensions@saveblocktag

```
3011 \def\eql@dimensions@saveblocktag{%
3012   \eql@dimensions@add{\eql@row@0\relax,%
3013     \eql@tagwidth@block@\the\eql@tagwidth@\relax
3014     \eql@tagheight@block@\the\ht\eql@tagbox@\relax
3015     \eql@tagdepth@block@\the\dp\eql@tagbox@\relax
3016     \eql@dimensions@addreg\eql@tagpos@shift@
3017     \let\noexpand\eql@tagpos@reserve\ifdefined\eql@tagpos@reserve
3018     \noexpand\eql@true\else\noexpand\eql@false\fi
3019   };}%
3020   \global\eql@tagwidth@max@\eql@tagwidth@
3021   \global\eql@tagrows@\@ne
3022 }
```

sions@savenoblocktag

```
3023 \def\eqldimensions@savenoblocktag{%  
3024 \eqldimensions@add{\eqldimensions@row@0\relax,;}%  
3025 }
```

\eqldimensions@for

```
3026 \def\eqldimensions@for#1{%  
3027 \def\eqldimensions@forall{#1}%  
3028 \expandafter\eqldimensions@forstep\eqldimensions@tab  
3029 }
```

\eqldimensions@forstep

```
3030 \def\eqldimensions@forstep\eqldimensions@row@#1\relax#2,#3;%  
3031 \eqldimensions@row@#1\relax  
3032 \ifnum\eqldimensions@row@=#2\else  
3033 #3%  
3034 \def\eqldimensions@cells{#2}%  
3035 \eqldimensions@forall  
3036 \expandafter\eqldimensions@forstep  
3037 \fi  
3038 }
```

\eqldimensions@get

```
3039 \def\eqldimensions@get#1{%  
3040 \eqldimensions@row@#1\relax  
3041 \expandafter\eqldimensions@getdef\expandafter{\the\eqldimensions@row@}%  
3042 \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil  
3043 }
```

\eqldimensions@getdef

```
3044 \def\eqldimensions@getdef#1{%  
3045 \def\eqldimensions@getparse  
3046 ##1\eqldimensions@row@#1\relax##2,##3;##4\@nil{%  
3047 ##3%  
3048 \def\eqldimensions@cells{##2}%  
3049 }%  
3050 }
```

\eqldimensions@colwidth@tab

```
3051 \let\eqldimensions@colwidth@tab\empty
```

\eqldimensions@colwidth@get

```
3052 \def\eqldimensions@colwidth@get#1{%  
3053 \ifcase\eqldimensions@colwidth@tab\else\z@\fi  
3054 }
```

\eqldimensions@colwidth@save

```
3055 \def\eqldimensions@colwidth@save#1{%  
3056 \edef\eqldimensions@colwidth@tab{%  
3057 \noexpand\or\the#1%  
3058 \unexpanded\expandafter{\eqldimensions@colwidth@tab}%  
3059 }%  
3060 }
```

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

```

3061 \def\eql@dimensions@calc{%
3062   \eql@column@\z@
3063   \eql@line@pos@\z@
3064   \eql@line@possep@\z@
3065   \eql@line@avail@\eql@totalwidth@
3066   \eql@line@availsep@\eql@intercolumns@
3067   \eql@line@width@\z@
3068   \eql@line@widthsep@\z@
3069   \let\eql@dimensions@cellcall\eql@dimensions@calc@call
3070   \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
3071   \eql@dimensions@cells
3072 }

```

`ensions@calc@callsep` Callback for each intercolumn space.

```

3073 \def\eql@dimensions@calc@callsep{%
3074   \advance\eql@line@possep@\@ne
3075 }%

```

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

```

3076 \def\eql@dimensions@calc@call{%
3077   \advance\eql@column@\@ne
3078   \ifnum\eql@totalcolumns@=\@ne
3079     \dimen@\eql@totalwidth@
3080   \else
3081     \dimen@\eql@colwidth@get\eql@column@\relax
3082   \fi
3083   \ifdim\eql@cellwidth@>\z@
3084     \ifdim\eql@line@width@=\z@
3085       \eql@line@avail@\eql@line@pos@
3086       \eql@line@availsep@\eql@line@possep@
3087       \ifcase\eql@shape@pos@
3088         \or
3089           \advance\eql@line@avail@\dimexpr
3090             (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
3091         \or
3092           \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
3093         \fi
3094       \advance\eql@line@avail@\eql@shape@amount@
3095     \fi
3096     \eql@line@width@\eql@line@pos@
3097     \eql@line@widthsep@\eql@line@possep@
3098     \ifcase\eql@shape@pos@
3099       \advance\eql@line@width@\eql@cellwidth@
3100     \or
3101       \advance\eql@line@width@\dimexpr
3102         (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
3103     \or
3104       \advance\eql@line@width@\dimen@

```

```

3105   \fi
3106   \advance\eq@line@width@\eq@shape@amount@
3107   \fi
3108   \advance\eq@line@pos@\dimen@
3109 }

```

9.4 Best Line Selection

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

```

3110 \let\eq@numbering@best@auto\eq@false

```

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

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

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

```

3111 \newcount\eq@numbering@best@row@
3112 \newdimen\eq@numbering@best@space@
3113 \let\eq@numbering@best@use\eq@false

```

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

```

3114 \def\eq@numbering@best@find{%
3115   \eq@numbering@best@row@ \z@
3116   \eq@numbering@best@space@ \z@
3117   \eq@dimensions@for{%
3118     \eq@dimensions@calc
3119     \ifdefined\eq@tagsleft
3120       \dimen@\eq@line@avail@
3121     \else
3122       \dimen@\dimexpr\eq@totalwidth-\eq@line@width\relax
3123     \fi
3124     \ifdim\dimen@>\eq@numbering@best@space@
3125       \eq@numbering@best@row@\eq@row@
3126       \eq@numbering@best@space@\dimen@
3127     \fi
3128   }%
3129   \ifnum\eq@numbering@best@row@>\z@
3130     \eq@tagpos@row@\eq@numbering@best@row@
3131     \let\eq@tagpos@continuous\eq@false
3132     \eq@tagpos@prevrow@\z@
3133   \fi
3134 }

```

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

```

3135 \def\eq@numbering@best@test#1{%
3136   \eq@dimensions@get#1%
3137   \eq@dimensions@calc
3138   \ifdefined\eq@tagsleft
3139     \dimen@\dimexpr\eq@line@avail@
3140       +\eq@marginleft+\eq@line@availsep\eq@colsep\relax
3141   \else
3142     \dimen@\dimexpr\displaywidth-\eq@line@width@
3143       -\eq@marginleft-\eq@line@widthsep\eq@colsep\relax
3144   \fi
3145   \ifdim\dimen@<\eq@tagwidth@block@
3146     \let\eq@numbering@best@use\eq@true
3147   \fi
3148 }

```

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

```
3149 \def\eql@numbering@best@eval{%
3150   \ifdefined\eql@numbering@best@auto
3151     \ifdefined\eql@numbering@best@use\else
3152       \ifdefined\eql@numbering@multi\else
3153         \ifnum\eql@tagpos@row@>\z@
3154           \eql@numbering@best@test\eql@tagpos@row@
3155         \fi
3156       \ifnum\eql@tagpos@prevrow@>\z@
3157         \eql@numbering@best@test\eql@tagpos@prevrow@
3158       \fi
3159     \fi
3160   \fi
3161 \fi
3162 \ifdefined\eql@numbering@best@use
3163   \eql@numbering@best@find
3164 \fi
3165 }
```

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.

`\adjust@calc@tagmargin`

```
3166 \def\eql@adjust@calc@tagmargin{%
3167   \ifdefined\eql@tagmargin@val
3168     \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
3169   \else
3170     \eql@tagmargin@\eql@tagwidth@max@
3171     \ifdim\eql@tagmargin@>\z@
3172       \advance\eql@tagmargin@-\eql@tagsepmin@
3173     \fi
3174   \fi

3175   \dimen@\eql@tagrows@\p@
3176   \ifnum\eql@totalrows@=\@one
3177     \ifnum\eql@tagrows@=\@one
3178       \advance\dimen@1sp\relax
3179     \fi
3180   \fi
3181   \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
3182     \eql@tagmargin@\z@
3183   \fi

3184   \@tempdima\dimexpr\displaywidth
3185     -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
3186   \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
3187   \ifdim\@tempdimb>\z@
3188     \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
3189       \eql@tagmargin@\z@
3190     \fi
3191   \fi
3192 }
```

9.6 Single Column

ql@adjust@calc@lines

```
3193 \def\eq@adjust@calc@lines{%
3194   \eq@totalcolumns@\ne
3195   \eq@intercolumns@\z@
3196   \eq@colsep@\z@
3197   \ifdefined\eq@layoutleft
3198     \eq@marginleft@\glueexpr\eq@layoutleftmargin\relax
3199     \eq@marginleft@min@\glueexpr\eq@layoutleftmarginmin\relax
3200     \ifdim\eq@marginleft@<\eq@marginleft@min@
3201       \eq@marginleft@\eq@marginleft@min@
3202     \fi
3203     \dimen@\glueexpr\eq@layoutleftmarginmax\relax
3204     \ifdim\eq@marginleft@>\dimen@
3205       \eq@marginleft@\dimen@
3206     \fi
3207     \eq@marginright@\z@
3208     \eq@centeroffset@\z@
3209   \else
3210     \eq@adjust@calc@tagmargin
3211     \ifdefined\eq@paddingleft@val
3212       \eq@marginleft@\dimexpr
3213         (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3214         -\glueexpr\eq@paddingleft@val\relax\relax
3215       \ifdim\eq@marginleft@<\z@
3216         \eq@marginleft@\z@
3217       \fi
3218     \else
3219       \eq@marginleft@\z@
3220     \fi
3221     \ifdefined\eq@paddingright@val
3222       \eq@marginright@\dimexpr
3223         (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3224         -\glueexpr\eq@paddingright@val\relax\relax
3225       \ifdim\eq@marginright@<\z@
3226         \eq@marginright@\z@
3227       \fi
3228     \else
3229       \eq@marginright@\z@
3230     \fi
3231     \ifdim\eq@tagmargin@>\z@
3232       \ifdefined\eq@tagsleft
3233         \ifdim\eq@marginleft@<\eq@tagsepmin@
3234           \eq@marginleft@\eq@tagsepmin@
3235         \fi
3236         \advance\eq@marginleft@\eq@tagmargin@
3237         \advance\eq@centeroffset@\eq@tagmargin@
3238       \else
3239         \ifdim\eq@marginright@<\eq@tagsepmin@
3240           \eq@marginright@\eq@tagsepmin@
3241         \fi
3242         \advance\eq@marginright@\eq@tagmargin@
3243         \advance\eq@centeroffset@-\eq@tagmargin@
3244       \fi
3245     \fi
3246     \eq@marginleft@min@\z@
3247     \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@
```

```

3248     \ifdefined\eql@tagsleft+\else-\fi\eql@tagmargin@relax
3249 \fi

3250 \eql@totalwidth@\dimexpr\displaywidth
3251     -\eql@marginleft@-\eql@marginright@relax
3252 }

```

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:

```

3253 \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:

```

3254 \ifnum\eql@totalcolumns@<\thr@@
3255     \eql@totalcolumns@\tw@
3256     \let\eql@columns@fulllength\eql@false
3257 \fi

```

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

```

3258 \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@relax

```

Evaluate the minimum intercolumn space which we will need often:

```

3259 \eql@colsepmin@\glueexpr\eql@colsepmin@valrelax

```

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

```

3260 \ifdefined\eql@layoutleft
3261     \eql@marginleft@\glueexpr\eql@layoutleftmargin@relax
3262     \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin@relax
3263     \ifdim\eql@marginleft@<\eql@marginleft@min@
3264         \eql@marginleft@\eql@marginleft@min@
3265     \fi
3266 \else

```

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

```

3267     \eql@adjust@calc@tagmargin
3268     \ifdefined\eql@columns@fulllength
3269         \ifdim\eql@tagmargin@>\z@
3270             \advance\eql@tagmargin@\eql@tagsepmin@
3271         \fi
3272     \fi
3273     \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
3274         -\eql@intercolumns@\eql@colsepmin@relax
3275     \eql@tagmargin@\z@
3276     \fi
3277     \eql@marginleft@min@\z@
3278 \fi

```

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

```
3279 \ifnum\eql@intercolumns@>\z@
```

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

```
3280 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
3281 \ifdefined\eql@layoutleft
3282 \advance\eql@colsep@-\eql@marginleft@
3283 \else
3284 \advance\eql@colsep@-\eql@tagmargin@
3285 \fi
3286 \count@\eql@intercolumns@
3287 \ifdefined\eql@columns@fulllength\else
3288 \ifdefined\eql@layoutleft
3289 \advance\count@\@ne
3290 \else
3291 \advance\count@\tw@
3292 \fi
3293 \fi
3294 \divide\eql@colsep@\count@
```

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

```
3295 \ifdim\eql@colsep@<\eql@colsepmin@
3296 \eql@colsep@\eql@colsepmin@
3297 \else
3298 \ifdefined\eql@columns@fulllength\else
3299 \dimen@\glueexpr\eql@colsepmax@val\relax
3300 \ifdim\eql@colsep@>\dimen@
3301 \eql@colsep@\dimen@
3302 \fi
3303 \fi
3304 \fi
3305 \else
```

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

```
3306 \eql@colsep@\eql@colsepmin@
3307 \fi
```

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

```
3308 \ifdefined\eql@layoutleft
```

Set the default value:

```
3309 \ifdim\eql@colsep@=\eql@colsepmin@
```

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

```
3310 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
3311 -\eql@intercolumns@\eql@colsep@\relax
3312 \ifdim\dimen@<\eql@marginleft@
3313 \ifdim\dimen@<\eql@marginleft@min@
3314 \eql@marginleft@\eql@marginleft@min@
3315 \else
```

```

3316         \eql@marginleft@\dimen@
3317     \fi
3318 \fi
3319 \fi
3320 \else

```

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

```

3321 \ifdefined\eql@columns@fulllength
3322     \eql@marginleft@z@

```

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

```

3323 \else
3324     \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@
3325         -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
3326     \ifdim\eql@marginleft@<z@
3327         \eql@marginleft@z@
3328     \fi
3329 \fi

```

Add tag margin in case of left tags:

```

3330 \ifdefined\eql@tagsleft
3331     \advance\eql@marginleft@\eql@tagmargin@
3332 \fi
3333 \fi

```

Find the best row for tag placement:

```

3334 \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:

```

3335 \ifdefined\eql@tagsleft
3336     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
3337 \else
3338     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
3339 \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

```

3340 \ifdefined\eql@numbering@multi
3341     \eql@dimensions@for{%
3342         \ifdim\eql@tagwidth@>z@
3343             \eql@dimensions@calc
3344             \eql@adjust@columns@test
3345         \fi
3346     }%
3347 \else
3348     \ifnum\eql@tagpos@row@>z@
3349         \ifnum\eql@tagpos@row@>\eql@totalrows@\else
3350             \eql@dimensions@get\eql@tagpos@row@
3351             \eql@tagwidth@\eql@tagwidth@block@
3352             \eql@dimensions@calc
3353             \eql@adjust@columns@test
3354         \fi

```

```

3355 \fi
3356 \ifnum\eqL@tagpos@prevrow>\z@
3357 \eqL@dimensions@get\eqL@tagpos@prevrow@
3358 \eqL@tagwidth@\eqL@tagwidth@block@
3359 \eqL@dimensions@calc
3360 \eqL@adjust@columns@test
3361 \fi
3362 \fi

```

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

```

3363 \advance\eqL@totalwidth@\dimexpr
3364 \eqL@intercolumns@\eqL@colsep@+\eqL@marginleft@\relax
3365 }

```

Placement for Right Tags.

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

```

3366 \def\eqL@adjust@columns@test@tagsright{%

```

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

```

3367 \@tempdima\dimexpr\displaywidth-\eqL@line@width@-\eqL@tagwidth@\relax

```

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

```

3368 \ifdim\@tempdima<\dimexpr
3369 \eqL@marginleft@+\eqL@line@widthsep@\eqL@colsep@\relax

```

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

```

3370 \ifdim\@tempdima<\dimexpr
3371 \eqL@marginleft@min@+\eqL@line@widthsep@\eqL@colsepmin@\relax\else

```

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

```

3372 \eqL@adjust@columns@modify@tagsright
3373 \fi
3374 \fi
3375 }

```

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

```

3376 \def\eqL@adjust@columns@modify@tagsright{%

```

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

```

3377 \ifnum\eqL@line@widthsep>\z@
3378 \dimen@\@tempdima
3379 \count@\eqL@line@widthsep@
3380 \ifdefined\eqL@layoutleft
3381 \advance\dimen@-\eqL@marginleft@

```

```

3382 \else
3383 \ifdefined\eql@columns@fulllength\else
3384 \advance\count@\@ne
3385 \fi
3386 \fi
3387 \divide\dimen@\count@

```

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

```

3388 \ifdim\dimen@<\eql@colsep@
3389 \ifdim\dimen@<\eql@colsepmin@
3390 \eql@colsep@\eql@colsepmin@
3391 \else
3392 \eql@colsep@\dimen@
3393 \fi
3394 \fi
3395 \fi

```

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

```

3396 \dimen@\dimexpr\@tempdima-\eql@line@widthsep@\eql@colsep@\relax
3397 \ifdim\eql@marginleft@>\dimen@
3398 \eql@marginleft@\dimen@
3399 \fi
3400 }

```

Placement for Left Tags.

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

```

3401 \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.

```

3402 \count@\numexpr\eql@intercolumns@-\eql@line@availsep@\relax
3403 \@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.

```

3404 \ifdim\@tempdima>\dimexpr
3405 \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).

```

3406 \ifdim\eql@tagwidth@<%
3407 \ifdefined\eql@layoutleft
3408 \glueexpr\eql@layoutleftmarginmax\relax
3409 \else
3410 \displaywidth
3411 \fi

```

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

```

3412 \ifdim\@tempdima>\dimexpr
3413 \displaywidth-\eql@totalwidth@-\count@\eql@colsepmin@\relax\else

```

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

```
3414     \eql@adjust@columns@modify@tagsleft
3415     \fi
3416     \fi
3417 \fi
3418 }
```

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

```
3419 \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.

```
3420 \ifnum\count@>\z@
3421   \dimen@dimexpr\displaywidth-\eql@totalwidth-\@tempdima\relax
3422   \ifdefined\eql@columns@fulllength\else
3423     \advance\count@\@ne
3424   \fi
3425   \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.

```
3426   \ifdim\dimen@<\eql@colsep@
3427     \ifdim\dimen@<\eql@colsepmin@
3428       \dimen@\eql@colsepmin@
3429     \fi
3430     \advance\dimen@-\eql@colsep@
3431     \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3432     \advance\eql@colsep@\dimen@
3433   \fi
3434 \fi
```

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

```
3435   \dimen@dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3436   \ifdim\eql@marginleft@<\dimen@
3437     \eql@marginleft@\dimen@
3438   \fi
3439 }
```

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:

```
3440 \ifdefined\inf@bad\else%
3441   \newcount\inf@bad
3442   \inf@bad1000000\relax
3443 \fi
```

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

```
3444 \let\eql@restore@hfuzz\@empty
3445 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

`\eql@alignbadness@` The registers `\eql@alignbadness@` and `\eql@tagbadness@` store the allowable badness `\eql@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:

```
3446 \newcount\eql@alignbadness@
3447 \newcount\eql@tagbadness@
3448 \newcount\eql@arrange@badness@
3449 \eql@alignbadness@\inf@bad
3450 \eql@tagbadness@\inf@bad
```

10.2 Arrangement Methods

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

```
3451 \def\eql@arrange@try#1{%
3452   \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax
3453     \setbox\eql@cellbox@\hbox to\displaywidth{%
3454       \unhbox\eql@cellbox@\unkern\kern#1}%
3455     \eql@arrange@badness@\badness
3456   \else
3457     \eql@arrange@badness@\m@ne
3458   \fi
3459 }
```

`\eql@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

```
3460 \def\eql@arrange@print#1#2{%
3461   \eql@restore@hfuzz
3462   \if@eqnsw
3463     \ifdefined\eql@tagsleft
3464       \eql@tagbox@print@tagsleft
3465     \fi
3466   \fi
3467   \hbox to\displaywidth{%
3468     #1%
3469     \unhbox\eql@cellbox@\unkern
3470     #2%
3471     \eql@tagging@mathaddlast
3472   }%
3473   \if@eqnsw
3474     \ifdefined\eql@tagsleft\else
3475       \eql@tagbox@print@tagsright
3476     \fi
3477   \fi
3478 }
```

`\eql@arrange@print@alignleft` Fit the current equation line with the selected alignment within a given left and right
`\eql@arrange@print@aligncenter`
`\eql@arrange@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:

```

3479 \def\eql@arrange@print@alignleft#1#2{%
3480   \eql@display@firstavail@set{\dimexpr#1\relax}%
3481   \eql@arrange@print{\kern#1}{\kern#2}%
3482 }

3483 \def\eql@arrange@print@alignright#1#2{%
3484   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3485   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3486 }

3487 \def\eql@arrange@print@aligncenter#1{%
3488   \eql@display@firstavail@set{\dimexpr
3489     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3490   \ifdim#1>\z@
3491     \eql@arrange@print{\kern#1\hfil}{}%
3492   \else
3493     \eql@arrange@print{\hfil}{\kern-#1}%
3494   \fi
3495 }

```

`\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:

```

3496 \def\eql@arrange@init{%
3497   \eql@save@hfuzz
3498   \hfuzz\maxdimen
3499   \eql@shape@select
3500 }

```

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

```

3501 \def\eql@arrange@print@line{%
3502   \eql@tagging@tagaddbox
3503   \csname eql@arrange%
3504     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3505     @init\endcsname
3506   \csname eql@arrange%
3507     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3508     @\ifdefined\eql@tagpos@reserve
3509       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3510       notag\fi\endcsname
3511 }

```

10.3 Central Alignment

TODO: describe

```

3512 \def\eql@arrange@aligncenter@init{%
3513   \eql@tagging@aligncenter
3514   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3515     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3516 }

```

TODO: describe

```
3517 \def\eq@arrange@aligncenter@notag{%
3518   \ifdim\dimexpr\displaywidth-\eq@cellwidth@relax>%
3519     \ifdim\eq@line@offset@<\eq@marginleft@min@
3520       \dimexpr\tw@eq@marginleft@min@-\eq@line@offset@relax
3521     \else
3522       \eq@line@offset@
3523     \fi
3524   \eq@arrange@print@aligncenter\eq@line@offset@
3525 \else
3526   \ifdim\eq@line@offset@<\eq@marginleft@min@
3527     \eq@arrange@print@alignleft\eq@marginleft@min@\z@
3528   \else
3529     \eq@arrange@print@alignright\eq@marginleft@min@\z@
3530   \fi
3531 \fi
3532 }
```

TODO: describe

```
3533 \def\eq@arrange@aligncenter@tagsright{%
3534   \ifdim\dimexpr\displaywidth-\eq@cellwidth@relax>%
3535     \ifdim\eq@line@offset@<\dimexpr\eq@marginleft@min@-\eq@tagwidth@relax
3536       \dimexpr\tw@eq@marginleft@min@-\eq@line@offset@relax
3537     \else
3538       \dimexpr\tw@eq@tagwidth@+\eq@line@offset@relax
3539     \fi
3540   \eq@arrange@print@aligncenter\eq@line@offset@
3541 \else
3542   \eq@arrange@try{\dimexpr\eq@tagwidth@+\eq@marginleft@min@relax}%
3543   \ifnum\eq@arrange@badness@<\eq@tagbadness@
3544     \ifdim\eq@line@offset@<\dimexpr\eq@marginleft@min@-\eq@tagwidth@relax
3545       \eq@arrange@print@alignleft\eq@marginleft@min@\eq@tagwidth@
3546     \else
3547       \eq@arrange@print@alignright\eq@marginleft@min@\eq@tagwidth@
3548     \fi
3549   \else
3550     \let\eq@tagpos@reserve\eq@false
3551     \eq@arrange@aligncenter@notag
3552   \fi
3553 \fi
3554 }
```

```
3555 \def\eq@arrange@aligncenter@tagsleft{%
3556   \ifdim\eq@tagwidth@>\eq@marginleft@min@
3557     \ifdim\dimexpr\displaywidth-\eq@cellwidth@relax>%
3558       \ifdim\eq@line@offset@<\eq@tagwidth@
3559         \dimexpr\tw@eq@tagwidth@-\eq@line@offset@relax
3560       \else
3561         \eq@line@offset@
3562       \fi
3563     \eq@arrange@print@aligncenter\eq@line@offset@
3564   \else
3565     \eq@arrange@try\eq@tagwidth@
3566     \ifnum\eq@arrange@badness@<\eq@tagbadness@
3567       \ifdim\eq@line@offset@<\eq@tagwidth@
3568         \eq@arrange@print@alignleft\eq@tagwidth@\z@
3569       \else
3570         \eq@arrange@print@alignright\eq@tagwidth@\z@

```

```

3571     \fi
3572     \else
3573         \let\eql@tagpos@reserve\eql@false
3574         \eql@arrange@aligncenter@notag
3575     \fi
3576 \fi
3577 \else
3578     \eql@arrange@aligncenter@notag
3579 \fi
3580 }

```

10.4 Left Alignment

```

3581 \def\eql@arrange@alignleft@init{%
3582     \eql@tagging@alignleft
3583     \eql@line@offset@\dimexpr\eql@marginleft@+\eql@shape@amount@\relax
3584     \ifdim\eql@line@offset@<\eql@marginleft@min@
3585         \eql@line@offset@\eql@marginleft@min@
3586     \fi
3587 }

3588 \def\eql@arrange@alignleft@notag{%
3589     \ifdim\eql@line@offset@>\eql@marginleft@min@
3590         \eql@arrange@try\eql@line@offset@
3591         \ifnum\eql@arrange@badness@<\eql@alignbadness@
3592             \eql@arrange@print@alignleft\eql@line@offset@\z@
3593         \else
3594             \eql@arrange@print@alignright\eql@marginleft@min@\z@
3595         \fi
3596     \else
3597         \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3598     \fi
3599 }

3600 \def\eql@arrange@alignleft@tagsright{%
3601     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3602     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3603         \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3604     \else
3605         \ifdim\eql@line@offset@>\eql@marginleft@min@
3606             \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3607         \fi
3608         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3609             \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3610         \else
3611             \let\eql@tagpos@reserve\eql@false
3612             \eql@arrange@alignleft@notag
3613         \fi
3614     \fi
3615 }

3616 \def\eql@arrange@alignleft@tagsleft{%
3617     \ifdim\eql@tagwidth@>\eql@marginleft@min@
3618         \ifdim\eql@line@offset@>\eql@tagwidth@
3619             \eql@arrange@try\eql@line@offset@
3620             \ifnum\eql@arrange@badness@<\eql@alignbadness@
3621                 \eql@arrange@print@alignleft\eql@line@offset@\z@
3622             \else
3623                 \eql@arrange@try\eql@tagwidth@
3624             \ifnum\eql@arrange@badness@<\eql@tagbadness@

```

```

3625     \eql@arrange@print@alignright\eql@tagwidth@z@
3626     \else
3627     \let\eql@tagpos@reserve\eql@false
3628     \eql@arrange@print@alignright\eql@marginleft@min@z@
3629     \fi
3630     \fi
3631     \else
3632     \eql@arrange@try\eql@tagwidth@
3633     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3634     \eql@arrange@print@alignleft\eql@tagwidth@z@
3635     \else
3636     \let\eql@tagpos@reserve\eql@false
3637     \eql@arrange@alignleft@notag
3638     \fi
3639     \fi
3640     \else
3641     \eql@arrange@alignleft@notag
3642     \fi
3643 }

```

10.5 Right Alignment

```

3644 \def\eql@arrange@alignright@init{%
3645   \eql@tagging@alignright
3646   \eql@line@offset@\dimexpr\eql@marginright@-\eql@shape@amount@relax
3647   \ifdim\eql@line@offset@<z@
3648     \eql@line@offset@z@
3649   \fi
3650 }

```

TODO: describe

```

3651 \def\eql@arrange@alignright@notag{%
3652   \ifdim\eql@line@offset@>z@
3653     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@relax}%
3654     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3655       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3656     \else
3657       \eql@arrange@print@alignleft\eql@marginleft@min@z@
3658     \fi
3659   \else
3660     \eql@arrange@print@alignright\eql@marginleft@min@z@
3661   \fi
3662 }

```

TODO: describe

```

3663 \def\eql@arrange@alignright@tagsright{%
3664   \ifdim\eql@line@offset@>\eql@tagwidth@
3665     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@relax}%
3666     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3667       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3668     \else
3669       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@relax}%
3670       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3671         \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3672       \else
3673         \let\eql@tagpos@reserve\eql@false
3674         \eql@arrange@print@alignleft\eql@marginleft@min@z@
3675       \fi
3676     \fi

```

```

3677 \else
3678 \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3679 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3680 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3681 \else
3682 \let\eql@tagpos@reserve\eql@false
3683 \eql@arrange@alignright@notag
3684 \fi
3685 \fi
3686 }

```

TODO: describe

```

3687 \def\eql@arrange@alignright@tagsleft{%
3688 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3689 \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3690 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3691 \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3692 \else
3693 \ifdim\eql@line@offset@>\z@
3694 \eql@arrange@try\eql@tagwidth@
3695 \fi
3696 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3697 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3698 \else
3699 \let\eql@tagpos@reserve\eql@false
3700 \eql@arrange@alignright@notag
3701 \fi
3702 \fi
3703 \else
3704 \eql@arrange@alignright@notag
3705 \fi
3706 }

```

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`

```

3707 \def\eql@box@cr{%
3708 \ifmmode\else\unskip\fi
3709 \eql@vspaceskip@\z@skip
3710 \let\eql@punct@term\eql@false
3711 \let\eql@class@rel@composed\@empty
3712 \eql@ampprotect\eql@box@cr@test\eql@box@cr@process
3713 }

```

TODO: describe

```

3714 \def\eq@box@cr@test@setopt{\let\eq@box@cr@test\eq@box@cr@testopt}
3715 \def\eq@box@cr@test@setall{\let\eq@box@cr@test\eq@box@cr@testall}

```

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

```

3716 \def\eq@box@cr@testopt#1{\eq@teststaropt@tight
3717   {\eq@box@cr@testopt@set{#1}}{\eq@box@cr@testopt@set{#1}}{Opt}}
3718 \def\eq@box@cr@testopt@set#1[#2]{\advance\eq@vspaceskip@glueexpr#2\relax#1}

```

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

box@cr@testall@parse

```

3719 \def\eq@box@cr@testall{\eq@parseopt@cr\eq@box@cr@testall@parse}
3720 \def\eq@box@cr@testall@parse{%
3721   \ifx\eq@parseopt@token[%
3722     \let\eq@parseopt@next\eq@parseopt@vspace
3723   \fi
3724   \ifx\eq@parseopt@token*%
3725     \let\eq@parseopt@next\eq@parseopt@gobble
3726   \fi
3727   \ifx\eq@parseopt@token.%
3728     \let\eq@parseopt@next\eq@parseopt@punctpass
3729   \fi
3730   \ifx\eq@parseopt@token,%
3731     \let\eq@parseopt@next\eq@parseopt@punctpass
3732   \fi
3733   \ifx\eq@parseopt@token~%
3734     \let\eq@parseopt@next\eq@parseopt@punctpass
3735   \fi
3736   \ifx\eq@parseopt@token'%
3737     \let\eq@parseopt@next\eq@parseopt@punctnext
3738   \fi
3739   \ifx\eq@parseopt@token!%
3740     \let\eq@parseopt@next\eq@parseopt@punctterm
3741   \fi
3742   \ifx\eq@parseopt@token/%
3743     \let\eq@parseopt@next\eq@parseopt@punctclear
3744   \fi
3745   \ifx\eq@parseopt@token=%
3746     \let\eq@parseopt@next\eq@parseopt@relsymp
3747   \fi
3748   \ifx\eq@parseopt@token;%
3749     \let\eq@parseopt@next\eq@parseopt@relcont
3750   \fi
3751   \ifx\eq@parseopt@token:%
3752     \let\eq@parseopt@next\eq@parseopt@relstart
3753   \fi
3754   \ifx\eq@parseopt@token|%
3755     \let\eq@parseopt@next\eq@parseopt@relord
3756   \fi
3757   \ifx\eq@parseopt@token?%
3758     \let\eq@parseopt@next\eq@box@cr@parse@rel
3759   \fi
3760   \ifx\eq@parseopt@token&%
3761     \let\eq@parseopt@next\eq@parseopt@end
3762   \fi
3763 }
3764 \def\eq@box@cr@parse@rel#1#2{%
3765   \def\eq@tmp{#2}%
3766   \ifx\eq@tmp\eq@relax\else

```

```

3767 \ifdefined\eqlopunct@next\else
3768 \eqlopunct@next@clear
3769 \fi
3770 \fi
3771 \ifx\eqlopbox@open\eqlopbox@lines@open
3772 \ifx\eqlop@tmp\@empty
3773 \def\eqlopclass@rel@composed{\eqlopshape@cont}%
3774 \else
3775 \def\eqlopclass@rel@composed{\eqlopshape@rel#2}%
3776 \fi
3777 \fi
3778 \ifx\eqlopbox@open\eqlopbox@columns@open
3779 \def\eqlopclass@rel@composed{\eqlopclass@rel@make{#2}}%
3780 \fi
3781 \ifx\eqlopbox@open\eqlopbox@cases@open
3782 \def\eqlopclass@rel@composed{&#2}%
3783 \fi
3784 \eqlop@parseopt@end}

```

`\eqlopbox@cr@process`

```

3785 \def\eqlopbox@cr@process{%
3786 \ifdefined\eqlopunct@term\eqlopunct@apply@top\fi
3787 \edef\eqlop@tmp{%
3788 \unexpanded{%
3789 \eqlopbox@endline
3790 \eqlopbox@lastcell
3791 \cr
3792 }%
3793 \noalign{%
3794 \vskip\the\eqlop@vspaceskip@relax
3795 }%
3796 \unexpanded\expandafter{\eqlopclass@rel@composed}%
3797 }%
3798 \eqlop@tmp
3799 }

```

`\eqlopbox@endline`

```

3800 \def\eqlopbox@endline{%
3801 \eqlopunct@apply@line
3802 \eqlop@hook@lineout
3803 }

```

11.2 Column Breaks

TODO: describe

```

3804 \def\eqlopbox@amp{%
3805 \eqlop@ampprotecttwo\eqlopbox@amp@testescape\eqlop@amp@org
3806 \eqlopbox@amp@process}
3807 \def\eqlopbox@amp@testescape#1#2{\eqlop@ifnextgobble@tight/{#1}{%
3808 \relax
3809 \let\eqlopunct@term\eqlop@false
3810 \let\eqlopclass@rel@composed\@empty
3811 \eqlopbox@amp@test{#2}}}

```

TODO: describe

```

3812 \def\eqL@box@amp@test@setopt{%
3813   \let\eqL@box@amp@test\eqL@box@amp@testopt}
3814 \def\eqL@box@amp@test@setall{%
3815   \let\eqL@box@amp@test\eqL@box@amp@testall}

```

\eqL@box@amp@testopt **TODO:** describe

```
3816 \let\eqL@box@amp@testopt\@empty
```

\eqL@box@amp@testall **TODO:** describe

ox@amp@testall@parse

```

3817 \def\eqL@box@amp@testall{\eqL@parseopt@cr\eqL@box@amp@testall@parse}
3818 \def\eqL@box@amp@testall@parse{%
3819   \ifx\eqL@parseopt@token.%
3820     \let\eqL@parseopt@next\eqL@parseopt@punctpass
3821   \fi
3822   \ifx\eqL@parseopt@token,%
3823     \let\eqL@parseopt@next\eqL@parseopt@punctpass
3824   \fi
3825   \ifx\eqL@parseopt@token~%
3826     \let\eqL@parseopt@next\eqL@parseopt@punctpass
3827   \fi
3828   \ifx\eqL@parseopt@token'%
3829     \let\eqL@parseopt@next\eqL@parseopt@punctnext
3830   \fi
3831   \ifx\eqL@parseopt@token!%
3832     \let\eqL@parseopt@next\eqL@parseopt@punctterm
3833   \fi
3834   \ifx\eqL@parseopt@token=%
3835     \let\eqL@parseopt@next\eqL@parseopt@relsymp
3836   \fi
3837   \ifx\eqL@parseopt@token;%
3838     \let\eqL@parseopt@next\eqL@parseopt@relcont
3839   \fi
3840   \ifx\eqL@parseopt@token:%
3841     \let\eqL@parseopt@next\eqL@parseopt@relstart
3842   \fi
3843   \ifx\eqL@parseopt@token|%
3844     \let\eqL@parseopt@next\eqL@parseopt@relord
3845   \fi
3846   \ifx\eqL@parseopt@token?%
3847     \let\eqL@parseopt@next\eqL@box@amp@parse@rel
3848   \fi
3849   \ifx\eqL@parseopt@token&%
3850     \let\eqL@parseopt@next\eqL@parseopt@end
3851   \fi
3852 }
3853 \def\eqL@box@amp@parse@rel#1#2{%
3854   \ifx\eqL@box@open\eqL@box@columns@open
3855     \def\eqL@class@rel@composed{\eqL@class@rel@make{#2}}%
3856   \else
3857     \def\eqL@class@rel@composed{#2}%
3858   \fi
3859   \eqL@parseopt@end}

```

\eqL@box@amp@process

```

3860 \def\eqL@box@amp@process{%
3861   \ifdefined\eqL@punct@term\eqL@punct@apply@top\fi

```

```

3862 \ifx\eql@box@open\eql@box@columns@open
3863 \edef\eql@tmp{%
3864 \ifx\eql@class@rel@composed\@empty
3865 \ifx\eql@box@lastcell\eql@box@columns@lastcell@odd
3866 &\noexpand\eql@punct@next@clear\fi&%
3867 \else
3868 \ifx\eql@box@lastcell\eql@box@columns@lastcell@even&\fi%
3869 \unexpanded\expandafter{\eql@class@rel@composed}%
3870 \fi
3871 }%
3872 \else
3873 \edef\eql@tmp{%
3874 \ifx\eql@class@rel@composed\@empty
3875 &%
3876 \else
3877 \unexpanded\expandafter{\eql@class@rel@composed}%
3878 \fi
3879 }%
3880 \fi
3881 \eql@tmp
3882 }

```

11.3 Lines Mode

```

3883 \def\eql@box@lines@lastcell{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3884 \def\eql@box@lines@open{%
3885 \eql@shape@align@enable
3886 \let\eql@box@lastcell\eql@box@lines@lastcell
3887 \everycr{\noalign{%
3888 \eql@verbose@info\eql@verbose@msg@startline@number
3889 \global\advance\eql@row@\@ne
3890 }}}%
3891 \tabskip\z@skip
3892 \halign\bgroup
3893 &%
3894 \global\let\eql@cell@container\@empty
3895 \setbox\eql@cellbox@\hbox{%
3896 \eql@strut@cell
3897 \@lign
3898 $\m@th\eql@mathstyle
3899 \eql@hook@cclin
3900 ##%
3901 \eql@punct@apply@col
3902 \eql@hook@colout
3903 \eql@tagging@mathsave
3904 $%
3905 \eql@tagging@mathaddlast
3906 }%
3907 \ifdefined\eql@shape@lastrow
3908 \eql@totalrows@\eql@row@
3909 \fi
3910 \eql@shape@eval
3911 \eql@cell@container
3912 \ifdefined\eql@frame@cmd
3913 \ifcase\eql@shape@pos@
3914 \eql@frame@measure

```

```

3915     \advance\eq@shape@amount@-\eq@frame@margin@
3916     \or\or
3917     \eq@frame@measure
3918     \advance\eq@shape@amount@+\eq@frame@margin@
3919     \fi
3920     \eq@frame@print
3921     \fi
3922     \ifcase\eq@shape@pos@
3923     \kern\eq@shape@amount@
3924     \box\eq@cellbox@
3925     \hskip\glueexpr\eq@paddingleft@+\eq@paddingright@
3926     -\eq@shape@amount@+\@flushglue\relax
3927     \eq@tagging@alignleft
3928     \or
3929     \hskip\glueexpr\eq@paddingleft@+\eq@shape@amount@+\@flushglue\relax
3930     \box\eq@cellbox@
3931     \hskip\glueexpr\eq@paddingright@-\eq@shape@amount@+\@flushglue\relax
3932     \eq@tagging@aligncenter
3933     \or
3934     \hskip\glueexpr\eq@paddingleft@+\eq@paddingright@
3935     +\eq@shape@amount@+\@flushglue\relax
3936     \box\eq@cellbox@
3937     \kern-\eq@shape@amount@
3938     \eq@tagging@alignright
3939     \fi
3940     \tabskip\eq@colsep@\relax
3941     \crrc
3942     \noalign{%
3943     \global\let\eq@shape@lastrow\eq@false
3944     \eq@hook@blockbefore
3945     }%
3946     \eq@hook@blockin
3947 }
3948 \def\eq@box@lines@set{\let\eq@box@open\eq@box@lines@open}

```

11.4 Columns Mode

```

3949 \def\eq@box@columns@lastcell@odd{%
3950   &\omit
3951   \eq@prevwidth@\wd\eq@cellbox@
3952   \let\eq@frame@cmd\eq@frame@prevcmd
3953   \ifdefined\eq@frame@cmd
3954     \eq@frame@measure
3955     \advance\eq@prevwidth@\eq@frame@margin@
3956     \eq@frame@print
3957   \fi
3958   \kern-\eq@prevwidth@
3959   \unhbox\eq@cellbox@
3960   \hfil
3961   &\omit\kern-\eq@colsep@
3962 }%
3963 \def\eq@box@columns@lastcell@even{&\omit\kern-\eq@colsep@}
3964 \def\eq@box@columns@open{%
3965 % \TODO templates
3966 \eq@shape@align@disable
3967 \let\eq@box@lastcell@empty
3968 \everycr{\noalign{%
3969   \eq@verbose@info\eq@verbose@msg@startline@new

```

```

3970 }}%
3971 \tabskip\z@skip
3972 \halign\bgroup
3973   &%
3974   \let\eq@box@lastcell\eq@box@columns@lastcell@odd
3975   \global\let\eq@cell@container\@empty
3976   \global\setbox\eq@cellbox@\hbox{%
3977     \eq@strut@cell
3978     \@lign
3979     $\m@th\eq@mathstyle
3980     \eq@hook@colin
3981     ##%
3982     \eq@punct@apply@next
3983     \eq@class@innerleft
3984     \eq@hook@innerleft
3985     \eq@tagging@mathsave
3986     $%
3987     \eq@tagging@mathaddlast
3988   }%
3989   \eq@cell@container
3990   \hfil
3991   \kern\wd\eq@cellbox@
3992   \ifdefined\eq@frame@cmd
3993     \eq@frame@measure
3994     \kern\eq@frame@margin@
3995   \fi
3996   \global\let\eq@frame@prevcmd\eq@frame@cmd
3997   \tabskip\z@skip
3998   &%
3999   \eq@prevwidth@\wd\eq@cellbox@
4000   \let\eq@box@lastcell\eq@box@columns@lastcell@even
4001   \let\eq@frame@cmd\eq@frame@prevcmd
4002   \global\let\eq@cell@container\@empty
4003   \setbox\eq@cellbox@\hbox{%
4004     \unhbox\eq@cellbox@
4005     \eq@strut@cell
4006     \@lign
4007     $\m@th\eq@mathstyle
4008     \eq@hook@innerright
4009     \eq@class@innerright@sel
4010     ##%
4011     \eq@punct@apply@col
4012     \eq@hook@colout
4013     \eq@tagging@mathsave
4014     $%
4015     \eq@tagging@mathaddlast
4016   }%
4017   \eq@cell@container
4018   \ifdefined\eq@frame@cmd
4019     \eq@frame@measure
4020     \advance\eq@prevwidth@\eq@frame@margin@
4021     \eq@frame@print
4022   \fi
4023   \kern-\eq@prevwidth@
4024   \unhbox\eq@cellbox@
4025   \hfil
4026   \tabskip\eq@colsep@\relax
4027 \crcr

```

```

4028 \noalign{%
4029 \eql@hook@blockbefore
4030 }%
4031 \eql@hook@blockin
4032 }
4033 \def\eql@box@columns@set{\let\eql@box@open\eql@box@columns@open}

```

11.5 Cases Mode

TODO: describe

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

```

4034 \def\eql@box@cases@lastcell{&}%

4035 \let\eql@box@cases@condtext\eql@false
4036 \let\eql@box@cases@condintro\@empty

4037 \def\eql@box@cases@open{%
4038 \eql@shape@align@disable
4039 \let\eql@box@lastcell\@empty
4040 \everycr{\noalign{%
4041 \eql@verbose@info\eql@verbose@msg@startline@new
4042 }}%
4043 \tabskip\z@skip
4044 \halign\bgroup
4045 \let\eql@box@lastcell\eql@box@cases@lastcell
4046 \global\let\eql@cell@container\@empty
4047 \global\setbox\eql@cellbox@\hbox{%
4048 \eql@strut@cell
4049 \@lign
4050 $\m@th\eql@mathstyle
4051 \eql@hook@colin
4052 ##%
4053 \eql@punct@apply@col
4054 \eql@tagging@mathsave
4055 $%
4056 \eql@tagging@mathaddlast
4057 }%
4058 \eql@cell@container
4059 \unhbox\eql@cellbox@
4060 \hfil
4061 \eql@tagging@alignleft
4062 \tabskip\eql@colsep@\relax
4063 &%
4064 \let\eql@box@lastcell\@empty
4065 \global\let\eql@cell@container\@empty
4066 \setbox\eql@cellbox@\hbox{%
4067 \unhbox\eql@cellbox@
4068 \eql@strut@cell
4069 \@lign
4070 $\m@th\eql@mathstyle
4071 \ifdefined\eql@box@cases@condtext
4072 \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
4073 \eql@box@cases@condintro
4074 ##%
4075 \eql@punct@apply@col
4076 \egroup

```

```

4077     \eql@hook@colout
4078     \eql@tagging@mathsave
4079     $%
4080     \eql@tagging@mathaddlast
4081     }%
4082     \eql@cell@container
4083     \unhbox\eql@cellbox@
4084     \hfil
4085     \eql@tagging@alignleft
4086     \tabskip\z@skip
4087     \crrc
4088     \noalign{%
4089         \eql@hook@blockbefore
4090     }%
4091     \eql@hook@blockin
4092 }

4093 \def\eql@box@cases@set{%
4094     \ifdefined\eql@cases@mathstyle\let\eql@mathstyle\eql@cases@mathstyle\fi
4095     \ifdefined\eql@punct@cases\let\eql@punct@col\eql@punct@cases\fi
4096     \let\eql@box@open\eql@box@cases@open}

```

11.6 Main

```

4097 \let\eql@box@box\vcenter
4098 \let\eql@box@open@undefined
4099 \let\eql@box@frame@firstofone
4100 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}
4101 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4102     \left#1##1\right#2}}
4103 \def\eql@box@getdim{\setbox@one\null\ht@one\ht\z@\dp@one\dp\z@}
4104 \def\eql@box@delim#1{\hbox{$\m@th\null\delimiterspace\z@\left#1%
4105     \ifx\eql@box@box\vcenter\vcenter{\box@one}\else\box@one\fi\right.$}}
4106 \def\eql@box@ldelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4107     \eql@box@getdim\eql@box@delim#1##1}}
4108 \def\eql@box@rdelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4109     \eql@box@getdim##1\eql@box@delim#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!

```

4110 \def\eql@box@close{%
4111     \ifvmode\else
4112         \ifmmode\else\unskip\fi
4113         \global\let\eql@shape@lastrow\eql@true
4114         \eql@punct@apply@block
4115         \ifdefined\eql@box@punct@term
4116             \eql@punct@apply@top
4117         \fi
4118         \eql@box@endline
4119         \eql@box@lastcell
4120         \cr
4121     \fi
4122     \noalign{%
4123         \eql@hook@blockafter

```

```

4124     \global\let\eq@shape@lastrow\eq@false
4125   }%
4126   \eq@tagging@tablesaveinner
4127 \egroup
4128 }

```

`\eq@box@vcenter`

```

4129 \def\eq@box@vcenter#1{%
4130   \ifmmode
4131     \vcenter{#1}%
4132   \else
4133     $\m@th\vcenter{#1}$%
4134   \fi
4135 }

```

`\eq@box@start`

```

4136 \let\eq@box@endmath\eq@false
4137 \def\eq@box@start{%
4138   \relax
4139   \ifmmode
4140     \let\eq@box@endmath\eq@false
4141   \else
4142     \let\eq@box@endmath\eq@true
4143     \expandafter$%$
4144   \fi
4145   \eq@beamerbasecolor@avoid
4146   \eq@stack@save@box
4147   \let\eq@frame@cmd\undefined
4148   \let\eq@layoutleft\eq@false
4149   \eq@row@z@
4150   \eq@totalrows@M
4151   \eq@shape@select
4152   \setbox\z@\ifx\eq@box@box\vcenter
4153     \expandafter\vbox
4154   \else
4155     \expandafter\eq@box@box
4156   \fi\bgroup
4157   \let\eqnpunct\eq@punct@setnext
4158   \eq@display@nest
4159   \let\\eq@box@cr
4160   \ifdefined\eq@amp@mode
4161     \let\&\eq@box@amp
4162   \fi
4163   \eq@spread@set
4164   \eq@strut@make
4165   \eq@box@open
4166 }

```

`\eq@box@end`

```

4167 \def\eq@box@end{%
4168   \eq@box@close
4169   \egroup
4170   \eq@box@framef%
4171   \ifdefined\eq@display@marginleft
4172     \hskip\glueexpr\eq@display@marginleft\relax
4173   \fi

```

```

4174 \ifx\eql@box@box\vcenter
4175 \eql@box@vcenter{\unvbox\z@}%
4176 \else
4177 \box\z@
4178 \fi
4179 \eql@tagging@tableaddinner
4180 \ifdefined\eql@display@marginright
4181 \hskip\glueexpr\eql@display@marginright\relax
4182 \fi
4183 }%
4184 \eql@stack@restore
4185 \ifdefined\eql@box@endmath
4186 \expandafter$$
4187 \fi
4188 }

```

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

```

4189 \def\eql@box@main{%
4190 \eql@box@start
4191 \eql@scan@body
4192 \eql@box@end
4193 }

```

11.7 Options Processing

TODO: describe

```

4194 \def\eql@box@test@setopt{\let\eql@box@test\eql@box@testopt}
4195 \def\eql@box@test@setall{\let\eql@box@test\eql@box@testall}

```

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

```

4196 \def\eql@box@testopt#1{\eql@testopt@tight{\eql@box@testopt@set{#1}}{}}
4197 \def\eql@box@testopt@set#1[#2]{\eqnaddopt{#2}#1}

```

```

4198 \def\eql@box@testall{\eql@parseopt@env\eql@box@testall@parse}
4199 \def\eql@box@testall@parse{%
4200 \ifx\eql@parseopt@token[%]
4201 \let\eql@parseopt@next\eql@parseopt@opt
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@punctpass
4211 \fi
4212 \ifx\eql@parseopt@token'%
4213 \let\eql@parseopt@next\eql@parseopt@punctopt
4214 \fi
4215 \ifx\eql@parseopt@token!%
4216 \let\eql@parseopt@next\eql@box@parse@punctterm
4217 \fi
4218 \ifx\eql@parseopt@token=%
4219 \let\eql@parseopt@next\eql@parseopt@lines

```

```

4220 \fi
4221 \ifx\eql@parseopt@token|%
4222   \let\eql@parseopt@next\eql@parseopt@columns
4223 \fi
4224 \ifx\eql@parseopt@token<%
4225   \let\eql@parseopt@next\eql@parseopt@ampeq
4226 \fi
4227 \ifx\eql@parseopt@token>%
4228   \let\eql@parseopt@next\eql@parseopt@eqamp
4229 \fi
4230 }
4231 \def\eql@box@parse@punctterm#1{\eqnadopt{punctterm}\eql@parseopt@peek}

```

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

```

4232 \def\eql@box@end@testall{\eql@parseopt@env\eql@box@end@testall@parse}
4233 \def\eql@box@end@testall@parse{%
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@punctpass
4242   \fi
4243   \ifx\eql@parseopt@token' %
4244     \let\eql@parseopt@next\eql@parseopt@punctblock
4245   \fi
4246   \ifx\eql@parseopt@token!%
4247     \let\eql@parseopt@next\eql@box@end@parse@punctterm
4248   \fi
4249   \ifx\eql@parseopt@token$%$
4250     \let\eql@parseopt@next\eql@parseopt@end
4251   \fi
4252 }
4253 \def\eql@box@end@parse@punctterm#1{%
4254   \let\eql@box@punct@term\eql@true\eql@parseopt@peek}

```

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

```

4255 \def\eql@box@processopt{%
4256   \let\eql@box@frame\@firstofone
4257   \let\eql@display@marginleft\@undefined
4258   \let\eql@display@marginright\@undefined
4259   \let\eql@box@punct@term\eql@false
4260   \let\eql@punct@block\@undefined
4261   \eql@nextopt@process{box}%
4262   \let\eql@punct@next\@undefined
4263   \eql@colsep@\glueexpr\eql@box@colsep\relax
4264   \ifdefined\eql@paddingleft@val
4265     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
4266   \else
4267     \eql@paddingleft@\z@
4268   \fi
4269   \ifdefined\eql@paddingright@val
4270     \eql@paddingright@\glueexpr\eql@paddingright@val\relax
4271   \else
4272     \eql@paddingright@\z@

```

```

4273 \fi
4274 \eql@indent@\glueexpr\eql@indent@val\relax
4275 }

```

11.8 Environment

`equationsbox` (*env.*)

```

4276 \newenvironment{equationsbox}{%
4277 \eql@verbose@info\eql@verbose@msg@enterenv
4278 \ifdefined\eql@box@env@modifier
4279 \eql@box@test@setall
4280 \else
4281 \eql@box@nomodifier
4282 \fi
4283 \eql@ampprotect\eql@box@test\eql@box@env@start
4284 }{%
4285 \ifdefined\eql@box@doscan\else
4286 \expandafter\eql@box@end
4287 \fi
4288 \eql@verbose@info\eql@verbose@msg@leaveenv
4289 }

```

`\eql@box@env@start`

```

4290 \def\eql@box@env@start{%
4291 \eql@box@processopt
4292 \ifdefined\eql@box@doscan
4293 \eql@box@call@set
4294 \expandafter\eql@scan@env
4295 \else
4296 \expandafter\eql@box@start
4297 \fi
4298 }

```

`\eql@box@call`

```

4299 \def\eql@box@call{\eql@box@main\eql@scan@end}
4300 \def\eql@box@call@test{%
4301 \eql@ampprotect\eql@box@end@testall\eql@box@call}
4302 \def\eql@box@call@set{%
4303 \ifdefined\eql@box@end@modifier
4304 \let\eql@scan@call\eql@box@call@test
4305 \else
4306 \let\eql@scan@call\eql@box@call
4307 \fi
4308 }

```

`\eql@box@ang@open`

```

4309 \newenvironment{equationsbox@ang}{-}{-}
4310 \def\eql@box@ang@open{%
4311 \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4312 \begin{equationsbox@ang}%
4313 \eql@verbose@info\eql@verbose@msg@enterenv
4314 \let\>\eql@box@ang@close
4315 \ifdefined\eql@box@ang@modifier
4316 \eql@box@test@setall

```

```

4317 \else
4318   \eql@box@nomodifier
4319 \fi
4320 \eql@ampprotect\eql@box@test\eql@box@ang@start
4321 }

```

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

```

4322 \def\eql@box@ang@start{%
4323   \eql@box@processopt
4324   \ifdefined\eql@box@dscan
4325     \eql@box@call@set
4326     \expandafter\eql@scan@ang
4327 \else
4328   \expandafter\eql@box@start
4329 \fi
4330 }

```

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

```

4331 \def\eql@box@ang@close{%
4332   \ifdefined\eql@box@dscan
4333     \let\eql@box@end@modifier\eql@false
4334 \fi
4335 \ifdefined\eql@box@end@modifier
4336   \expandafter\eql@ampprotect\expandafter\eql@box@end@testall
4337 \fi
4338 \eql@box@ang@end
4339 }

```

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

```

4340 \def\eql@box@ang@end{%
4341   \ifdefined\eql@box@dscan\else
4342     \expandafter\eql@box@end
4343 \fi
4344 \eql@verbose@info\eql@verbose@msg@leaveenv
4345 \end{equationsbox@ang}%
4346 \ignorespaces
4347 }

```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

4348 \def\eql@single@start@native{%
4349   \eql@display@init
4350   \eql@display@print
4351   \let\raisetag\eql@raisetag@default
4352   \eql@shape@align@disable
4353   \eql@hook@eqin
4354 }%

```

TODO: describe

```

4355 \def\eql@single@end@native{%
4356   \eql@tags@container
4357   \eql@numbering@single@eval
4358   \if@eqnsw
4359     \ifdefined\eql@tagsleft
4360       \leqno
4361     \else
4362       \eqno
4363     \fi
4364     \eql@composetag@print
4365   \fi
4366   \eql@interline@container
4367   \advance\eql@belowspace@\eql@vspaceskip@
4368   \eql@display@container
4369   \eql@display@penalty
4370   \eql@display@vspace@native
4371 }%

```

12.2 Print

```

4372 \def\eql@single@start@print{%
4373   \eql@display@init
4374   \eql@display@print
4375   \eql@shape@align@enable

4376   \eql@totalrows@\@ne
4377   \eql@row@\@ne
4378   \eql@arrange@init
4379   \global\let\eql@cell@container\@empty

4380   \prevgraf\numexpr\prevgraf+\@ne\relax
4381   \setbox\eql@cellbox@\hbox\bgroup
4382     \eql@restore@hfuzz
4383     \eql@strut@cell
4384     $\m@th\eql@mathstyle%$
4385     \eql@hook@eqin
4386 }

4387 \def\eql@single@end@print{%
4388   \eql@tagging@mathsave
4389   $%$
4390   \hfil
4391   \kern\z@
4392   \egroup
4393   \prevgraf\numexpr\prevgraf-\@ne\relax

4394   \eql@shape@eval
4395   \eql@cell@container

4396   \ifdefined\eql@frame@cmd
4397     \eql@frame@adjust
4398   \fi

4399   \eql@cellwidth@\wd\eql@cellbox@
4400   \eql@line@height@\ht\eql@cellbox@
4401   \eql@line@depth@\dp\eql@cellbox@
4402   \eql@totalwidth@\eql@cellwidth@
4403   \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
4404   \eql@topheight@\eql@line@height@
4405   \eql@bottomdepth@\eql@line@depth@

4406   \eql@tags@container
4407   \eql@numbering@single@eval

```

```

4408 \if@eqnsw
4409   \eql@tagbox@make\eql@composetag@print
4410   \eql@tagrows@\@ne
4411   \ifdefined\eql@tagpos@reserve\else
4412     \eql@tagwidth@\z@
4413   \fi
4414   \eql@tagheight@block@\ht\eql@tagbox@
4415   \eql@tagdepth@block@\dp\eql@tagbox@
4416 \else
4417   \eql@numbering@warnunused
4418   \eql@tagwidth@\z@
4419   \eql@tagrows@\z@
4420 \fi
4421 \eql@tagwidth@max@\eql@tagwidth@
4422 \eql@tagpos@single@eval
4423 \eql@tagpos@print@line@eval
4424 \eql@intercolumns@\z@
4425 \eql@adjust@calc@lines
4426 \eql@display@halign@init{ }%
4427 \halign{##\crr
4428   \noalign{\eql@display@halign@start}%
4429   \eql@arrange@print@line
4430   \cr
4431   \noalign{\eql@display@halign@end}%
4432   \eql@tagging@tablesavelines
4433 }%
4434 \eql@tagpos@print@line@end
4435 \eql@display@close
4436 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```

4437 \def\eql@lines@measure@line@begin{%
4438   \eql@verbose@info\eql@verbose@msg@startline@number
4439   \eql@numbering@measure@line@begin
4440   \eql@hook@linein
4441 }

```

TODO: describe

```

4442 \def\eql@lines@measure@line@end{%
4443   \eql@punct@apply@line
4444   \eql@hook@lineout
4445 }

```

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

```

4446 \def\eql@lines@measure@cell{%
4447   \ifdefined\eql@frame@cmd
4448     \ifcase\eql@shape@pos@

```

```

4449     \eql@frame@measure
4450     \advance\eql@shape@amount@-\eql@frame@margin@
4451     \or\or
4452     \eql@frame@measure
4453     \advance\eql@shape@amount@+\eql@frame@margin@
4454     \fi
4455     \eql@frame@print
4456     \fi
4457     \eql@cellwidth@\wd\eql@cellbox@
4458     \eql@line@height@\ht\eql@cellbox@
4459     \eql@line@depth@\dp\eql@cellbox@
4460     \eql@dimensions@startrow
4461     \eql@dimensions@savecell
4462     \kern\eql@cellwidth@
4463 }

```

`\eql@lines@measure`

```

4464 \def\eql@lines@measure{%
4465   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
4466   \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
4467   \ifdefined\eql@amp@mode
4468     \let&\eql@break@amp
4469   \fi
4470   \eql@totalrows@\@M
4471   \eql@shape@select

4472   \setbox\z@\vbox{\measuring@true\halign{%
4473     \global\let\eql@cell@container\@empty
4474     \setbox\eql@cellbox@\hbox{%
4475       \eql@strut@cell
4476       \@lign
4477       $\m@th\eql@mathstyle
4478       \eql@hook@colin
4479       ##%
4480       \eql@punct@apply@col
4481       \eql@hook@colout
4482       $%
4483     }%
4484     \ifdefined\eql@shape@lastrow
4485       \eql@totalrows@\eql@row@
4486     \fi
4487     \eql@shape@eval
4488     \eql@cell@container
4489     \eql@lines@measure@cell
4490     \eql@measure@tag
4491     \eql@measure@endrow
4492   \crrc

4493   \noalign{%
4494     \global\let\eql@shape@lastrow\eql@false
4495     \eql@hook@blockbefore
4496   }%
4497   \eql@hook@blockin
4498   \eql@scan@body
4499   \ifvmode\else
4500     \global\let\eql@shape@lastrow\eql@true
4501     \eql@punct@apply@block
4502     \eql@hook@blockout

```

```

4503     \eql@multi@endline
4504     \cr
4505     \fi
4506     \omit
4507     \cr
4508     \noalign{%
4509         \eql@hook@blockafter
4510         \global\let\eql@shape@lastrow\eql@false
4511     }%
4512 }}%

4513 \eql@measure@close

4514 \setbox\z@\vbox{%
4515     \unvbox\z@
4516     \unpenalty
4517     \global\setbox\@ne\lastbox
4518 }%
4519 \eql@totalwidth@\wd\@ne

4520 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@measure
4521 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

4522 \def\eql@lines@adjust{%
4523     \eql@tagpos@adjust@eval
4524     \eql@adjust@calc@lines
4525     \eql@numbering@best@eval
4526 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```

4527 \def\eql@lines@print@line@begin{%
4528     \eql@verbose@info\eql@verbose@msg@startline@number
4529     \eql@numbering@print@line@begin
4530     \eql@hook@linein
4531 }

```

TODO: describe

```

4532 \def\eql@lines@print@line@end{%
4533     \eql@punct@apply@line
4534     \eql@hook@lineout
4535 }

```

TODO: describe

```

4536 \def\eql@lines@print@line@adjust{%
4537     \ifdefined\eql@frame@cmd
4538         \ifcase\eql@shape@pos@
4539             \eql@frame@measure
4540             \advance\eql@shape@amount@-\eql@frame@margin@

```

```

4541 \or\or
4542 \eql@frame@measure
4543 \advance\eql@shape@amount@+\eql@frame@margin@
4544 \fi
4545 \eql@frame@adjust
4546 \fi
4547 \eql@cellwidth@\wd\eql@cellbox@
4548 \eql@line@height@\ht\eql@cellbox@
4549 \eql@line@depth@\dp\eql@cellbox@
4550 \eql@numbering@print@line@eval
4551 \if@eqnsw
4552 \eql@tagbox@make\eql@composetag@print
4553 \fi
4554 \eql@tagpos@print@line@eval
4555 \eql@arrange@print@line
4556 \eql@tagpos@print@line@end
4557 }

```

TODO: describe

```

4558 \def\eql@lines@print{%
4559 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@print
4560 \eql@arrange@init
4561 \eql@display@halign@init\eql@lines@print@line@begin
4562 \eql@multi@cr@let\eql@lines@print@line@end
4563 \ifdefined\eql@amp@mode
4564 \let&\eql@break@amp
4565 \fi
4566 \tabskip\z@skip

4567 \halign{%
4568 \global\let\eql@cell@container\@empty
4569 \setbox\eql@cellbox@\hbox{%
4570 \eql@restore@hfuzz
4571 \eql@strut@cell
4572 \@lign
4573 $\m@th\eql@mathstyle
4574 \eql@hook@colin
4575 ##%
4576 \eql@punct@apply@col
4577 \eql@hook@colout
4578 \eql@tagging@mathsave
4579 $%
4580 \hfil
4581 \kern\z@
4582 }%
4583 \eql@shape@eval
4584 \eql@cell@container
4585 \eql@lines@print@line@adjust
4586 \cr

4587 \noalign{%
4588 \eql@display@halign@start
4589 \eql@numbering@print@block@begin
4590 \eql@hook@blockbefore
4591 }%
4592 \eql@hook@blockin
4593 \eql@scan@body
4594 \ifvmode\else
4595 \relax

```



```

4632 \relax
4633 \let\eql@punct@term\eql@false
4634 \let\eql@class@rel@composed\@empty
4635 \eql@columns@amp@test{#2}}

```

TODO: describe

```

4636 \def\eql@columns@amp@test@setopt{%
4637 \let\eql@columns@amp@test\eql@columns@amp@testopt}
4638 \def\eql@columns@amp@test@setall{%
4639 \let\eql@columns@amp@test\eql@columns@amp@testall}

```

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

```

4640 \let\eql@columns@amp@testopt\@empty

```

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

ns@amp@testall@parse

```

4641 \def\eql@columns@amp@testall{\eql@parseopt@cr\eql@columns@amp@testall@parse}
4642 \def\eql@columns@amp@testall@parse{%
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@punctpass
4648 \fi
4649 \ifx\eql@parseopt@token~%
4650 \let\eql@parseopt@next\eql@parseopt@punctpass
4651 \fi
4652 \ifx\eql@parseopt@token'%
4653 \let\eql@parseopt@next\eql@parseopt@punctnext
4654 \fi
4655 \ifx\eql@parseopt@token!%
4656 \let\eql@parseopt@next\eql@parseopt@punctterm
4657 \fi
4658 \ifx\eql@parseopt@token=%
4659 \let\eql@parseopt@next\eql@parseopt@relsymp
4660 \fi
4661 \ifx\eql@parseopt@token;%
4662 \let\eql@parseopt@next\eql@parseopt@relcont
4663 \fi
4664 \ifx\eql@parseopt@token:%
4665 \let\eql@parseopt@next\eql@parseopt@relstart
4666 \fi
4667 \ifx\eql@parseopt@token|%
4668 \let\eql@parseopt@next\eql@parseopt@relord
4669 \fi
4670 \ifx\eql@parseopt@token?%
4671 \let\eql@parseopt@next\eql@columns@amp@parse@rel
4672 \fi
4673 \ifx\eql@parseopt@token&%
4674 \let\eql@parseopt@next\eql@parseopt@end
4675 \fi
4676 }
4677 \def\eql@columns@amp@parse@rel#1#2{%
4678 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
4679 \eql@parseopt@end}

```

@columns@amp@process

```

4680 \def\eql@columns@amp@process{%
4681   \ifdefined\eql@punct@term\eql@punct@apply@top\fi
4682   \edef\eql@tmp{%
4683     \ifx\eql@class@rel@composed\@empty
4684       \ifodd\eql@column@&\noexpand\eql@punct@next@clear\fi&%
4685     \else
4686       \ifodd\eql@column@\else&\fi%
4687       \unexpanded\expandafter{\eql@class@rel@composed}%
4688     \fi
4689   }%
4690   \eql@tmp
4691 }

```

14.3 Transpose

TODO: describe

TODO: adjust to \&?!

TODO: describe

```

4692 \let\eql@transpose@active\eql@false
4693 \def\eql@transpose@end{\eql@transpose@end}
4694 \def\eql@transpose@skip{\&\eql@punct@next@clear}
4695 \def\eql@transpose@complete{%
4696   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

TODO: describe

```

4697 \def\eql@transpose{%
4698   \eql@totalcolumns@z@
4699   \eql@totalrows@z@
4700   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@\&\eql@transpose@end\&%
4701   \eql@scan@reg@{}%
4702   \eql@row@z@
4703   \eql@transpose@output@row
4704 }

```

TODO: describe

```

4705 \def\eql@transpose@save@col#1{%
4706   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
4707     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

TODO: describe

```

4708 \def\eql@transpose@scan@col#1\&{%
4709   \def\eql@tmpa{#1}%
4710   \ifx\eql@tmpa\eql@transpose@end\else
4711     \advance\eql@totalcolumns@\@ne
4712     \eql@row@z@
4713     \let\eql@transpose@data@col\@empty
4714     \eql@transpose@scan@row#1\\eql@transpose@end\\%
4715     \ifnum\eql@row@>\eql@totalrows@
4716       \eql@totalrows@\eql@row@
4717     \fi
4718     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
4719     \expandafter\eql@transpose@scan@col
4720   \fi
4721 }

```

TODO: describe

```
4722 \def\eql@transpose@append@row#1{%
4723   \advance\eql@row@\@ne
4724   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}
```

TODO: describe

```
4725 \def\eql@transpose@scan@row#1\\{%
4726   \def\eql@tmpa{#1}%
4727   \ifx\eql@tmpa\eql@transpose@end\else
4728     \ifx\eql@transpose@active+
4729       \eql@transpose@scan@cell#1&\eql@transpose@end&%
4730     \else
4731       \eql@transpose@append@row{#1}%
4732     \fi
4733     \expandafter\eql@transpose@scan@row
4734   \fi
4735 }
```

TODO: describe

```
4736 \def\eql@transpose@scan@cell#1&#2&{%
4737   \def\eql@tmpa{#2}%
4738   \ifx\eql@tmpa\eql@transpose@end
4739     \eql@transpose@append@row{#1}%
4740   \else
4741     \eql@transpose@append@row{#1&#2}%
4742     \expandafter\eql@transpose@scan@cell@next
4743   \fi
4744 }
```

TODO: describe

```
4745 \def\eql@transpose@scan@cell@next#1&{%
4746   \def\eql@tmpa{#1}%
4747   \ifx\eql@tmpa\eql@transpose@end\else
4748     \eql@transpose@append@row{&#1}%
4749     \expandafter\eql@transpose@scan@cell@next
4750   \fi
4751 }
```

TODO: describe

```
4752 \def\eql@transpose@output@row{%
4753   \ifnum\eql@row@<\eql@totalrows@
4754     \advance\eql@row@\@ne
4755     \eql@column@\z@
4756     \eql@transpose@output@col
4757     \ifnum\eql@row@<\eql@totalrows@
4758       \eql@scan@addto\\%
4759     \fi
4760     \expandafter\eql@transpose@output@row
4761   \fi
4762 }
```

TODO: describe

```
4763 \def\eql@transpose@output@col{%
4764   \ifnum\eql@column@<\eql@totalcolumns@
4765     \advance\eql@column@\@ne
4766     \csname eql@transpose@data@col@the\eql@column@\endcsname
```

```

4767 \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4768 \ifnum\eql@column@<\eql@totalcolumns@
4769 \eql@scan@addto{\eql@transpose@complete}%
4770 \fi
4771 \expandafter\eql@transpose@output@col
4772 \fi
4773 }

```

14.4 Measure

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

s@measure@line@begin

```

4774 \def\eql@columns@measure@line@begin{%
4775 \eql@verbose@info\eql@verbose@msg@startline@number
4776 \global\eql@column@\z@
4777 \global\eql@line@height@\z@
4778 \global\eql@line@depth@\z@
4779 \eql@numbering@measure@line@begin
4780 \eql@hook@linein
4781 }

```

```

4782 \def\eql@columns@measure@cell{%
4783 \eql@cellwidth@\wd\eql@cellbox@
4784 \ifdefined\eql@frame@cmd
4785 \eql@frame@measure
4786 \advance\eql@cellwidth@\eql@frame@margin@
4787 \fi
4788 \ifdim\ht\eql@cellbox@>\eql@line@height@
4789 \global\eql@line@height@\ht\eql@cellbox@
4790 \fi
4791 \ifdim\dp\eql@cellbox@>\eql@line@depth@
4792 \global\eql@line@depth@\dp\eql@cellbox@
4793 \fi
4794 \ifnum\eql@column@=\@ne
4795 \eql@dimensions@startrow
4796 \fi
4797 \ifodd\eql@column@
4798 \eql@shape@pos@\tw@
4799 \else
4800 \eql@shape@pos@\z@
4801 \fi
4802 \eql@shape@amount@\z@
4803 \eql@dimensions@savecell
4804 \ifodd\eql@column@\else
4805 \eql@dimensions@savesep
4806 \fi
4807 \kern\eql@cellwidth@
4808 }

```

mns@measure@line@end

```

4809 \def\eql@columns@measure@line@end{%
4810 \eql@punct@apply@line
4811 \eql@hook@lineout
4812 &\omit
4813 \ifnum\eql@column@>\eql@totalcolumns@

```

```

4814 \global\eql@totalcolumns@eql@column@
4815 \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

```

4816 \ifdefined\eql@frame@cmd
4817 \advance\eql@column@\@ne
4818 \wd\eql@cellbox@z@
4819 \eql@columns@measure@cell
4820 \fi
4821 \eql@measure@tag
4822 \eql@measure@endrow
4823 }

```

`\eql@columns@measure`

```

4824 \def\eql@columns@measure{%
4825 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@measure
4826 \eql@totalcolumns@z@
4827 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end
4828 \ifdefined\eql@amp@mode
4829 \let\&\eql@columns@amp
4830 \fi

4831 \setboxz@\vbox{\measuring@true\halign{%
4832 &%
4833 \global\advance\eql@column@\@ne
4834 \global\let\eql@cell@container\@empty
4835 \global\setbox\eql@cellbox@\hbox{%
4836 \eql@strut@cell
4837 \@lign
4838 $\m@th\eql@mathstyle
4839 \eql@hook@colin
4840 ##%
4841 \eql@punct@apply@next
4842 \eql@class@innerleft
4843 \eql@hook@innerleft
4844 $%
4845 }%
4846 \eql@cell@container
4847 \hfil
4848 \eql@columns@measure@cell
4849 \global\let\eql@frame@prevcmd\eql@frame@cmd
4850 &%
4851 \eql@prevwidth@\wd\eql@cellbox@
4852 \let\eql@frame@cmd\eql@frame@prevcmd
4853 \global\advance\eql@column@\@ne
4854 \global\let\eql@cell@container\@empty
4855 \setbox\eql@cellbox@\hbox{%
4856 \eql@strut@cell
4857 \@lign
4858 $\m@th\eql@mathstyle
4859 \eql@hook@innerright
4860 \eql@class@innerright@sel
4861 ##%
4862 \eql@punct@apply@col
4863 \eql@hook@colout

```

```

4864     $%
4865     }%
4866     \eql@cell@container
4867     \eql@columns@measure@cell
4868     \hfil
4869     \crrc

4870     \noalign{%
4871         \eql@hook@blockbefore
4872     }%
4873     \eql@hook@blockin
4874     \eql@scan@body

4875     \ifvmode\else
4876         \eql@punct@apply@block
4877         \eql@hook@blockout
4878         \eql@multi@endline
4879         \cr
4880     \fi
4881     \noalign{%
4882         \eql@hook@blockafter
4883     }%

```

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

```

4884     \omit
4885     \eql@column@\@ne
4886     \eql@columns@completerow
4887     \cr
4888     }}%

4889     \eql@measure@close

4890     \setbox\z@\vbox{%
4891         \unvbox\z@
4892         \unpenalty
4893         \global\setbox\@ne\lastbox
4894     }%
4895     \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4896     \let\eql@colwidth@tab\@empty
4897     \loop
4898         \setbox\@ne\hbox{%
4899             \unhbox\@ne
4900             \unskip
4901             \global\setbox\thr@\@lastbox
4902         }%
4903     \ifhbox\thr@@
4904         \eql@colwidth@save{\wd\thr@@}%
4905     \repeat

4906     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@measure
4907 }

```

14.5 Columns Placement

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

```

4908 \def\eql@columns@adjust{%
4909   \ifodd\eql@totalcolumns@
4910     \advance\eql@totalcolumns@\@ne
4911   \fi
4912   \eql@tagpos@adjust@eval
4913   \eql@adjust@calc@columns
4914 }

```

14.6 Print

TODO: describe

ms@print@line@begin

```

4915 \def\eql@columns@print@line@begin{%
4916   \eql@verbose@info\eql@verbose@msg@startline@number
4917   \global\eql@column@\z@
4918   \global\eql@line@pos@\eql@marginleft@
4919   \global\eql@line@width@\z@
4920   \global\eql@line@avail@\eql@totalwidth@
4921   \global\eql@line@height@\z@
4922   \global\eql@line@depth@\z@
4923   \eql@numbering@print@line@begin
4924   \eql@hook@linein
4925 }

```

l@columns@print@cell

```

4926 \def\eql@columns@print@cell{%
4927   \eql@cellwidth@\wd\eql@cellbox@
4928   \ifodd\eql@column@
4929     \ifdefined\eql@frame@cmd
4930       \eql@frame@measure
4931       \advance\eql@cellwidth@\eql@frame@margin@
4932     \fi
4933     \dimen@\z@
4934   \else
4935     \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```

4936     \ifdefined\eql@frame@cmd
4937       \eql@frame@measure
4938       \advance\eql@cellwidth@\eql@frame@margin@
4939       \advance\eql@prevwidth@\eql@frame@margin@
4940       \eql@frame@print
4941     \fi

```

update height and depth

```

4942     \ifdim\ht\eql@cellbox@>\eql@line@height@
4943       \global\eql@line@height@\ht\eql@cellbox@
4944     \fi
4945     \ifdim\dp\eql@cellbox@>\eql@line@depth@
4946       \global\eql@line@depth@\dp\eql@cellbox@
4947     \fi

```

print box

```

4948 \kern-\eql@prevwidth@
4949 \unhbox\eql@cellbox@
4950 \dimen@-\eql@cellwidth@
4951 \fi

```

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

```

4952 \advance\dimen@\eql@colwidth@get\eql@column@\relax
4953 \kern\dimen@

```

update available and used space

```

4954 \dimen@\eql@colwidth@get\eql@column@\relax
4955 \ifdim\eql@cellwidth@>\z@
4956 \ifdim\eql@line@width@=\z@
4957 \eql@line@avail@\eql@line@pos@
4958 \ifodd\eql@column@
4959 \advance\eql@line@avail@\dimen@
4960 \advance\eql@line@avail@-\eql@cellwidth@
4961 \fi
4962 \global\eql@line@avail@\eql@line@avail@
4963 \fi
4964 \eql@line@width@\eql@line@pos@
4965 \ifodd\eql@column@
4966 \advance\eql@line@width@\dimen@
4967 \else
4968 \advance\eql@line@width@\eql@cellwidth@
4969 \fi
4970 \global\eql@line@width@\eql@line@width@
4971 \fi
4972 \advance\eql@line@pos@\dimen@
4973 \ifodd\eql@column@\else
4974 \advance\eql@line@pos@\eql@colsep@
4975 \fi
4976 \global\eql@line@pos@\eql@line@pos@
4977 }

```

```

4978 \def\eql@columns@print@trailright{%
4979 &\omit
4980 \eql@prevwidth@\wd\eql@cellbox@
4981 \let\eql@frame@cmd\eql@frame@prevcmd
4982 \global\advance\eql@column@\@ne
4983 \eql@columns@print@cell
4984 }

```

lums@print@line@end **TODO:** add an even column with empty stuff if box processing deferred

```

4985 \def\eql@columns@print@line@end{%
4986 \eql@punct@apply@line
4987 \eql@hook@lineout
4988 \ifodd\eql@column@
4989 \expandafter\eql@columns@print@trailright
4990 \fi
4991 \eql@columns@completerow
4992 \eql@columns@print@tag
4993 }

```

ql@columns@print@tag

```

4994 \def\eql@columns@print@tag{%
4995   \kern-\dimexpr\eql@totalwidth+\eql@colsep@relax

```

determine first line available space

```

4996   \eql@display@firstavail@set\eql@line@avail@
4997   \eql@columns@overfull
4998   \eql@numbering@print@line@eval
4999   \if@eqnsw
5000     \eql@tagbox@make\eql@composetag@print
5001   \fi
5002   \eql@tagpos@print@line@eval
5003   \eql@tagbox@print@cell
5004   \eql@tagpos@print@line@end
5005 }

```

\eql@columns@print

```

5006 \def\eql@columns@print{%
5007   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
5008   \eql@shape@align@disable
5009   \eql@display@halign@init\eql@columns@print@line@begin
5010   \eql@multi@cr@let\eql@columns@print@line@end
5011   \ifdefined\eql@amp@mode
5012     \let\&\eql@columns@amp
5013   \fi
5014   \tabskip\eql@marginleft@

5015   \halign{%
5016     &%
5017     \global\advance\eql@column@\@ne
5018     \global\let\eql@cell@container\@empty
5019     \global\setbox\eql@cellbox@\hbox{%
5020       \eql@strut@cell
5021       \@lign
5022       $\m@th\eql@mathstyle
5023       \eql@hook@colin
5024       ##%
5025       \eql@punct@apply@next
5026       \eql@class@innerleft
5027       \eql@hook@innerleft
5028       \eql@tagging@mathsave
5029       $%
5030       \eql@tagging@mathaddlast
5031     }%
5032     \eql@cell@container
5033     \hfil
5034     \eql@columns@print@cell
5035     \global\let\eql@frame@prevcmd\eql@frame@cmd
5036     \tabskip\z@skip
5037     &%
5038     \eql@prevwidth@\wd\eql@cellbox@
5039     \let\eql@frame@cmd\eql@frame@prevcmd
5040     \global\advance\eql@column@\@ne
5041     \global\let\eql@cell@container\@empty
5042     \setbox\eql@cellbox@\hbox{%
5043       \unhbox\eql@cellbox@
5044       \eql@strut@cell
5045       \@lign
5046       $\m@th\eql@mathstyle

```

```

5047     \eql@hook@innerright
5048     \eql@class@innerright@sel
5049     ##%
5050     \eql@punct@apply@col
5051     \eql@hook@colout
5052     \eql@tagging@mathsave
5053     $%
5054     \eql@tagging@mathaddlast
5055     }%
5056     \eql@cell@container
5057     \eql@columns@print@cell
5058     \hfil
5059     \tabskip\eql@colsep@\relax
5060 \crrc

5061 \noalign{%
5062     \eql@display@halign@start
5063     \eql@numbering@print@block@begin
5064     \eql@hook@blockbefore
5065     }%
5066     \eql@hook@blockin
5067     \eql@scan@body
5068     \ifvmode\else
5069         \relax
5070         \eql@punct@apply@block
5071         \eql@hook@blockout
5072         \eql@multi@endline
5073         \cr
5074     \fi
5075     \noalign{%
5076         \eql@hook@blockafter
5077         \eql@display@halign@end
5078         \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
5079     }%
5080     \eql@tagging@tablesavealign
5081 }%
5082 }

```

15 Interface

15.1 Options Processing

TODO: describe

```

5083 \def\eql@equations@test@setopt{\let\eql@equations@test\eql@equations@testopt}
5084 \def\eql@equations@test@setall{\let\eql@equations@test\eql@equations@testall}

```

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

```

5085 \def\eql@equations@testopt#1{%
5086     \eql@testopt@tight{\eql@equations@testopt@set{#1}}{}}
5087 \def\eql@equations@testopt@set#1[#2]{\eqnadopt{#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 `\eqnadopt`. All arguments are scanned such that any spaces stop the scanning and such that any

alignment markers '&' cannot interfere: **TODO:** update

```
5088 \def\eq@equations@testall{\eq@parseopt@env\eq@equations@testall@parse}
5089 \def\eq@equations@testall@parse{%
5090   \ifx\eq@parseopt@token*%
5091     \let\eq@parseopt@next\eq@parseopt@nonumber
5092   \fi
5093   \ifx\eq@parseopt@token!%
5094     \let\eq@parseopt@next\eq@parseopt@donumber
5095   \fi
5096   \ifx\eq@parseopt@token/%
5097     \let\eq@parseopt@next\eq@parseopt@transpose
5098   \fi
5099   \ifx\eq@parseopt@token[%]
5100     \let\eq@parseopt@next\eq@parseopt@opt
5101   \fi
5102   \ifx\eq@parseopt@token\eq@atxi
5103     \let\eq@parseopt@next\eq@parseopt@label
5104   \fi
5105   \ifx\eq@parseopt@token\eq@atxii
5106     \let\eq@parseopt@next\eq@parseopt@label
5107   \fi
5108   \ifx\eq@parseopt@token.%
5109     \let\eq@parseopt@next\eq@parseopt@punctpass
5110   \fi
5111   \ifx\eq@parseopt@token,%
5112     \let\eq@parseopt@next\eq@parseopt@punctpass
5113   \fi
5114   \ifx\eq@parseopt@token~%
5115     \let\eq@parseopt@next\eq@parseopt@punctpass
5116   \fi
5117   \ifx\eq@parseopt@token'%
5118     \let\eq@parseopt@next\eq@parseopt@punctopt
5119   \fi
5120   \ifx\eq@parseopt@token-%
5121     \let\eq@parseopt@next\eq@parseopt@single
5122   \fi
5123   \ifx\eq@parseopt@token=%
5124     \let\eq@parseopt@next\eq@parseopt@lines
5125   \fi
5126   \ifx\eq@parseopt@token|%
5127     \let\eq@parseopt@next\eq@parseopt@columns
5128   \fi
5129   \ifx\eq@parseopt@token<%
5130     \let\eq@parseopt@next\eq@parseopt@ampeq
5131   \fi
5132   \ifx\eq@parseopt@token>%
5133     \let\eq@parseopt@next\eq@parseopt@eqamp
5134   \fi
5135   \ifx\eq@parseopt@token\label
5136     \let\eq@parseopt@next\eq@parseopt@end
5137   \fi
5138   \ifx\eq@parseopt@token\begin
5139     \let\eq@parseopt@next\eq@parseopt@end
5140   \fi
5141 }
```

quations@end@testall **TODO:** describe

```

5142 \def\eqlequations@end@testall{%
5143   \eql@parseopt@env\eqlequations@end@testall@parse}
5144 \def\eqlequations@end@testall@parse{%
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@punctpass
5150   \fi
5151   \ifx\eql@parseopt@token~%
5152     \let\eql@parseopt@next\eql@parseopt@punctpass
5153   \fi
5154   \ifx\eql@parseopt@token'%
5155     \let\eql@parseopt@next\eql@parseopt@punctblock
5156   \fi
5157 }

```

`\eqlequations@processopt` The macro `\eqlequations@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:

```

5158 \def\eqlequations@processopt{%
5159   \let\eql@tags@container@block\eql@tags@container@clear
5160   \let\eql@tags@frame@cmd\@firstofone
5161   \let\eql@skip@force@above\@undefined
5162   \let\eql@skip@force@below\@undefined
5163   \let\eql@skip@force@leave\@undefined
5164   \let\eql@display@linewidth\@undefined
5165   \let\eql@display@marginleft\@undefined
5166   \let\eql@display@marginright\@undefined
5167   \eql@abovespace@\z@skip
5168   \eql@belowspace@\z@skip
5169   \eql@displaybreak@prepen@\@MM
5170   \eql@displaybreak@postpen@\@MM
5171   \eql@nextopt@process{equations}%
5172   \let\eql@punct@next\@undefined
5173   \eql@indent@\glueexpr\eql@indent@val\relax
5174   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
5175 }

```

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:

```

5176 \def\eql@single@cr@error{%
5177   \eql@error{Cannot use '\string\\' within display equation.
5178     Please switch to equations environment}%
5179 }

```

`\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:

```

5180 \def\eq@single@start{%
5181   \eq@display@enter
5182   \eq@tagging@start
5183   \eq@dollar@dollar@begin
5184   \eq@display@adjust
5185   \eq@numbering@init
5186   \eq@stack@save@equations
5187   \eq@numbering@single@init
5188   \ifdefined\eq@single@cr@mode
5189     \let\\\eq@single@cr@mode
5190   \fi
5191   \ifdefined\eq@amp@mode
5192     \let&\eq@break@amp
5193   \fi
5194   \ifdefined\eq@single@native
5195     \let\eq@single@start@sel\eq@single@start@native
5196     \let\eq@single@end@sel\eq@single@end@native
5197   \else
5198     \let\eq@single@start@sel\eq@single@start@print
5199     \let\eq@single@end@sel\eq@single@end@print
5200   \fi
5201   \eq@single@start@sel
5202 }

```

`\eq@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:

```

5203 \def\eq@single@end{%
5204   \eq@punct@apply@block
5205   \eq@hook@eqout
5206   \eq@single@end@sel
5207   \global\eq@punct@top@reset
5208   \eq@stack@restore
5209   \eq@dollar@dollar@end
5210   \eq@tagging@end
5211   \eq@display@leave
5212 }

```

`\eq@single@main` Combined opening, body and closing for pre-scanned body:

```

5213 \def\eq@single@main{%
5214   \eq@single@start
5215   \eq@scan@body
5216   \eq@single@end
5217 }

```

`equations@single@set` Configure equations macros to single-line mode:

```

5218 \def\eq@equations@single@set{%
5219   \ifdefined\eq@single@doscan
5220     \let\eq@equations@main\eq@single@main
5221   \else
5222     \let\eq@equations@main\@undefined
5223   \fi
5224 }

```

15.3 Multi-Line Main

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

```
5225 \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

```
5226 \def\eql@multi@main{%
5227   \eql@display@enter
5228   \eql@tagging@start
5229   \eql@dollar@dollar@begin
5230   \eql@display@adjust
5231   \eql@numbering@init
5232   \eql@stack@save@equations
5233   \ifdefined\eql@transpose@active
5234     \ifdefined\eql@multi@linesmode\else
5235       \eql@transpose
5236     \fi
5237   \fi
5238   \ifdefined\eql@numbering@subeq@use
5239     \eql@numbering@subeq@init
5240   \fi
5241   \eql@display@init
5242   \let\intertext\eql@intertext
5243   \let\endintertext\endeql@intertext
5244   \eql@shape@align@enable
```

Now measure the given multi-line equations body:

```
5245   \ifdefined\eql@multi@linesmode
5246     \eql@lines@measure
5247   \else
5248     \ifdefined\eql@ampproof@active
5249       \eql@ampproof
5250     \fi
5251     \eql@columns@measure
5252   \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`:

```
5253   \ifdefined\eql@numbering@subeq@use
5254     \eql@numbering@subeq@test
5255   \fi
5256   \ifdefined\eql@multi@linesmode\else
5257     \ifdefined\eql@multi@linesfallback
5258       \ifnum\eql@totalcolumns@=\@ne
5259         \let\eql@multi@linesmode\eql@true
5260       \ifx\eql@multi@linesfallback\z@\else
5261         \eql@lines@measure
5262       \fi
5263     \fi
5264   \fi
5265 \fi
```

Adjust the multi-line equations body:

```
5266 \ifdefined\eql@multi@linesmode
5267   \eql@lines@adjust
5268 \else
5269   \eql@columns@adjust
5270 \fi
```

Now print the multi-line equations body:

```
5271 \eql@display@print
5272 \eql@numbering@print@init
5273 \ifdefined\eql@multi@linesmode
5274   \eql@lines@print
5275 \else
5276   \eql@columns@print
5277 \fi
5278 \eql@display@close
```

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

```
5279 \ifdefined\eql@numbering@subeq@use
5280   \eql@numbering@subeq@close
5281 \fi
5282 \global\eql@punct@top@reset
5283 \eql@stack@restore
5284 \eql@dollar@dollar@end
5285 \eql@tagging@end
5286 \eql@display@leave
5287 }
```

`equations@columns@set` Configure equations macros to one of the two multi-line modes:
`@equations@lines@set`

```
5288 \def\eql@equations@columns@set{%
5289   \let\eql@equations@main\eql@multi@main
5290   \let\eql@multi@linesmode\eql@false
5291 }
5292 \def\eql@equations@lines@set{%
5293   \let\eql@equations@main\eql@multi@main
5294   \let\eql@multi@linesmode\eql@true
5295 }
```

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`:

```
5296 \newenvironment{equations}{%
5297   \ifmmode
5298     \expandafter\eql@equations@env@cancel
5299   \else
5300     \eql@verbose@info\eql@verbose@msg@enterenv
```

```

5301 \expandafter\eqlequations@env@open
5302 \fi
5303 }{
5304 \ifdefined\eqlequations@main\else
5305 \expandafter\eqlsingle@end
5306 \fi
5307 \ignorespacesafterend
5308 \eql@verbose@info\eql@verbose@msg@leaveenv
5309 }
5310 \eql@markline@amsthm@register{equations}
5311 \eql@tagging@register@luamml{equations}

```

equations@env@cancel

```

5312 \def\eqlequations@env@cancel{%
5313 \eql@error@mathmode{\string\begin{\@currentenv}}%
5314 \let\eql@scan@call\eql@scan@env@cancel
5315 \eql@scan@env
5316 }

```

lequations@env@open

```

5317 \def\eqlequations@env@open{%
5318 \ifdefined\eqlequations@env@modifier
5319 \eqlequations@test@setall
5320 \else
5321 \eqlequations@nomodifier
5322 \fi
5323 \eql@ampprotect\eqlequations@test\eqlequations@env@start
5324 }

```

@equations@env@start The macro `\eqlequations@env@start` first processes the arguments. Depending on the chosen mode of operation, scan the environment body passing on to `\eqlequations@main` or process a single-line equation via `\eqlsingle@start`:

```

5325 \def\eqlequations@env@start{%
5326 \eqlequations@processopt
5327 \ifdefined\eqlequations@main
5328 \eqlequations@call@set
5329 \expandafter\eql@scan@env
5330 \else
5331 \expandafter\eqlsingle@start
5332 \fi
5333 }

```

\eqlequations@call

```

5334 \def\eqlequations@call{\eqlequations@main\eql@scan@end}
5335 \def\eqlequations@call@test{%
5336 \eql@ampprotect\eqlequations@end@testall\eqlequations@call}
5337 \def\eqlequations@call@set{%
5338 \ifdefined\eqlequations@end@modifier
5339 \let\eql@scan@call\eqlequations@call@test
5340 \else
5341 \let\eql@scan@call\eqlequations@call
5342 \fi
5343 }

```

Square Brackets.

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

```
5344 \newenvironment{equations@sqr}{}{}
5345 \eql@markline@amsthm@register{equations@sqr}
5346 \eql@tagging@register@luamml{equations@sqr}
```

`\@equations@sqr@open` Definition for ‘`\[`’. Add the default arguments `\eql@equations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eql@equations@sqr@start`:

```
5347 \def\eql@equations@sqr@open{%
5348   \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
5349   \begin{equations@sqr}%
5350   \eql@verbose@info\eql@verbose@msg@enterenv
5351   \let\]\eql@equations@sqr@close
5352   \ifdefined\eql@equations@sqr@modifier
5353     \eql@equations@test@setall
5354   \else
5355     \eql@equations@nomodifier
5356   \fi
5357   \eql@ampprotect\eql@equations@test\eql@equations@sqr@start
5358 }
```

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

```
5359 \def\eql@equations@sqr@start{%
5360   \eql@equations@processopt
5361   \ifdefined\eql@equations@main
5362     \eql@equations@call@set
5363     \expandafter\eql@scan@sqr
5364   \else
5365     \expandafter\eql@single@start
5366   \fi
5367 }
```

`\@equations@sqr@close` Definition for ‘`\]`’. Parse modifiers following ‘`\]`’ and hand on to `\eql@equations@sqr@end`:

```
5368 \protected\def\eql@equations@sqr@close{%
5369   \ifdefined\eql@equations@main
5370     \let\eql@equations@end@modifier\eql@false
5371   \fi
5372   \ifdefined\eql@equations@end@modifier
5373     \expandafter\eql@ampprotect\expandafter\eql@equations@end@testall
5374   \fi
5375   \eql@equations@sqr@end
5376 }
```

`\@equations@sqr@end` **TODO:** complete End `\[...]` block:

```
5377 \def\eql@equations@sqr@end{%
5378   \ifdefined\eql@equations@main\else
5379     \expandafter\eql@single@end
5380   \fi
5381   \eql@verbose@info\eql@verbose@msg@leaveenv
```

```

5382 \end{equations@sq}
5383 \ignorespaces
5384 }

```

Angle Brackets.

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

```

5385 \newenvironment{equations@ang}{}{}
5386 \eql@markline@amsthm@register{equations@ang}
5387 \eql@tagging@register@luamml{equations@ang}

```

`l@equations@ang@open`

```

5388 \def\eql@equations@ang@open{%
5389 \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
5390 \begin{equations@ang}%
5391 \eql@verbose@info\eql@verbose@msg@enterenv
5392 \let\>\eql@equations@ang@close
5393 \ifdefined\eql@equations@ang@modifier
5394 \eql@equations@test@setall
5395 \else
5396 \eql@equations@nomodifier
5397 \fi
5398 \eql@ampprotect\eql@equations@test\eql@equations@ang@start
5399 }

```

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

```

5400 \def\eql@equations@ang@start{%
5401 \eql@equations@processopt
5402 \ifdefined\eql@equations@main
5403 \eql@equations@call@set
5404 \expandafter\eql@scan@ang
5405 \else
5406 \expandafter\eql@single@start
5407 \fi
5408 }

```

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

```

5409 \def\eql@equations@ang@close{%
5410 \ifdefined\eql@equations@main
5411 \let\eql@equations@end@modifier\eql@false
5412 \fi
5413 \ifdefined\eql@equations@end@modifier
5414 \expandafter\eql@ampprotect\expandafter\eql@equations@end@testall
5415 \fi
5416 \eql@equations@ang@end
5417 }

```

`ql@equations@ang@end` **TODO:** describe

```

5418 \def\eql@equations@ang@end{%
5419 \ifdefined\eql@equations@main\else
5420 \expandafter\eql@single@end
5421 \fi
5422 \eql@verbose@info\eql@verbose@msg@leaveenv

```

```

5423 \end{equations@ang}%
5424 \ignorespaces
5425 }

```

16 Options

16.1 Selection Tools

`\eql@decide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```

5426 \def\eql@decide@abovebelow#1#2#3#4#5{%
5427 \eql@decide@select{#1}{#2}{#3}{%
5428 {abovebelow,both,tb}{#4#5},%
5429 {above,top,t}{#4},%
5430 {below,bottom,b}{#5}}

```

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

```

5431 \def\eql@decide@situation#1#2#3#4{%
5432 \eql@decide@select{#1}{#2}{#3}{%
5433 {{long}{\def#4{0}}},%
5434 {{short}{\def#4{1}}},%
5435 {{cont}{\def#4{2}}},%
5436 {{par}{\def#4{3}}},%
5437 {{top}{\def#4{4}}},%
5438 {{noskip}{\def#4{5}}},%
5439 {{medskip}{\def#4{6}}}}

```

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

```

5440 \def\eql@decide@delim#1#2#3{%
5441 \eql@decide@select{#1}{#2}{#3}{%
5442 {{,.,\eql@decide@false}{\eql@box@wrap{}{}}},%
5443 {{\eql@decide@true,r,p,round,parentheses}{\eql@box@delim()}},%
5444 {{s,b,sqr,square}{\eql@box@delim[]}},%
5445 {{c,B,curly,braces}{\eql@box@delim\lbrace\rbrace}},%
5446 {{a,ang,angle}{\eql@box@delim\langle\rangle}},%
5447 {{v,vert}{\eql@box@delim\vert\vert}},%
5448 {{d,V,dvert}{\eql@box@delim\Vert\Vert}},%
5449 {\relax{\eql@box@delim#3}}}%

```

TODO: describe

```

5450 \def\eql@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:

```

5451 \eql@define@key{box}{lines,ln,gathered,gather,ga}[]{%
5452 \eql@box@lines@set}
5453 \eql@define@key{box}{columns,col,aligned,align,al}[]{%

```

```

5454 \eql@box@columns@set}
5455 \eql@define@key{box}{cases}[]{%
5456 \eql@box@cases@set\eql@box@ldelim\lbrace%
5457 \def\eql@box@colsep{\eql@box@condsep}}
5458 \eql@define@key{box}{matrix}[r]{%
5459 \eql@box@lines@set\eql@shape@set{center}%
5460 \let\eql@spread@reset\eql@true\def\eql@spread@val{\z0}%
5461 \def\eql@box@colsep{\eql@box@matrixsep}%
5462 \let\eql@mathstyle\@empty
5463 \eql@punct@clear
5464 \eql@box@cr@test@setopt
5465 \eql@box@amp@test@setopt
5466 \let\eql@box@end@modifier\eql@false
5467 \eql@decide@delim{#3}{#2}{#1}}
5468 \eql@define@key{box}{top,t}[]{\let\eql@box@box\vtop}
5469 \eql@define@key{box}{center,c}[]{\let\eql@box@box\vcenter}
5470 \eql@define@key{box}{bottom,b}[]{\let\eql@box@box\vbox}
5471 \eql@define@key{box}{intro}{%
5472 \def\eql@box@cases@condintro{#1}}
5473 \eql@define@key{box}{introtext}{%
5474 \def\eql@box@cases@condintro{%
5475 \ifmmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}
5476 \eql@define@key{box}{textcond}[true]{%
5477 \eql@decide@select{#3}{#2}{#1}{%
5478 {\eql@decide@true,text}{\let\eql@box@cases@condtext\eql@true}},%
5479 {\eql@decide@false,math}{\let\eql@box@cases@condtext\eql@false}}}}
5480 \eql@define@key{setup}{scanbox}[true]{%
5481 \eql@decide@bool{#3}{#2}{#1}\eql@box@doscan}
5482 \eql@define@key{box}{scan}[true]{%
5483 \eql@decide@bool{#3}{#2}{#1}\eql@box@doscan}
5484 \eql@define@key{setup}{boxangopt}[]{%
5485 \def\eql@box@ang@opt{columns,#1}}

```

Modes for Equations Environment.

```
5486 \let\eql@box@doscan\eqn@false
```

Declare modes and switches for the equations environment:

```

5487 \eql@define@key{equations}{single,1,equation,eq}[]{\eql@equations@single@set}
5488 \eql@define@key{equations}{lines,ln,gathered,gather,ga}[]{%
5489 \eql@equations@lines@set}
5490 \eql@define@key{equations}{columns,col,aligned,align,al}[]{%
5491 \eql@equations@columns@set}
5492 \eql@define@key{equations,setup}{transpose}[true]{%
5493 \eql@decide@select{#3}{#2}{#1}{%
5494 {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
5495 {\noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
5496 {\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}}
5497 \eql@define@key{setup}{nativeequation}[true]{%
5498 \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
5499 \eql@define@key{equations}{native}[true]{%
5500 \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
5501 \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
5502 \eql@define@key{setup}{scanequation}[true]{%
5503 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
5504 \eql@define@key{equations}{scan}[true]{%
5505 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
5506 \eql@define@key{setup}{sqropt}[]{%

```

```

5507 \def\eqlequations@sqr@opt{equation,#1}
5508 \eql@define@key{setup}{angopt}[]{%
5509 \def\eqlequations@ang@opt{columns,#1}}

```

TODO: describe

```

5510 \eql@define@key{control}{restoreexterior}[]{\eqldisplay@restore}
5511 \eql@define@key{control}{restoreexterior*}[]{\@arrayparboxrestore}

```

Modes for Modifier Scanning.

```

5512 \def\eqlequations@nomodifier{%
5513 \eqlequations@test@setopt
5514 \let\eqlequations@end@modifier\eql@false
5515 \eql@multi@cr@test@setopt
5516 \eql@columns@amp@test@setopt
5517 \eql@break@cr@test@setopt
5518 \eql@break@amp@test@setopt
5519 }
5520 \let\eqlequations@env@modifier\eql@false
5521 \let\eqlequations@sqr@modifier\eql@true
5522 \let\eqlequations@ang@modifier\eql@true
5523 \let\eqlequations@end@modifier\eql@true
5524 \eqlequations@test@setall
5525 \eql@multi@cr@test@setall
5526 \eql@columns@amp@test@setall
5527 \eql@break@cr@test@setall
5528 \eql@break@amp@test@setall
5529 \eql@break@test@setopt
5530 \let\eql@multi@cr@relnext\eql@true

5531 \def\eql@box@nomodifier{%
5532 \eql@box@test@setopt
5533 \let\eql@box@end@modifier\eql@false
5534 \eql@box@cr@test@setopt
5535 \eql@box@amp@test@setopt
5536 }
5537 \let\eql@box@env@modifier\eql@false
5538 \let\eql@box@ang@modifier\eql@true
5539 \let\eql@box@end@modifier\eql@true
5540 \eql@box@test@setall
5541 \eql@box@cr@test@setall
5542 \eql@box@amp@test@setall

5543 \eql@define@key{setup}{modifier}[true]{%
5544 \eql@decide@if{#3}{#2}{#1}%
5545   {\let\eqlequations@env@modifier\eql@true
5546     \let\eqlequations@sqr@modifier\eql@true
5547     \let\eqlequations@ang@modifier\eql@true
5548     \let\eql@box@env@modifier\eql@true
5549     \let\eql@box@ang@modifier\eql@true
5550     \eql@multi@cr@test@setall\eql@break@cr@test@setall
5551     \eql@columns@amp@test@setall\eql@break@amp@test@setall
5552     \eql@box@cr@test@setall\eql@box@amp@test@setall
5553     \eql@break@test@setall}}%
5554   {\let\eqlequations@env@modifier\eql@false
5555     \let\eqlequations@sqr@modifier\eql@false
5556     \let\eqlequations@ang@modifier\eql@false
5557     \let\eql@box@env@modifier\eql@false

```

```

5558 \let\eq@box@ang@modifier\eq@false
5559 \eq@multi@cr@test@setopt\eq@break@cr@test@setopt
5560 \eq@columns@amp@test@setopt\eq@break@amp@test@setopt
5561 \eq@box@cr@test@setopt\eq@box@amp@test@setopt
5562 \eq@break@test@setopt}}
5563 \eq@define@key{setup}{modifierenv}[true]{%
5564 \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier
5565 \eq@decide@bool{#3}{#2}{#1}\eq@box@env@modifier}
5566 \eq@define@key{setup}{modifiersqrang}[true]{%
5567 \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier
5568 \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier
5569 \eq@decide@bool{#3}{#2}{#1}\eq@box@ang@modifier}
5570 \eq@define@key{setup}{modifierend}[true]{%
5571 \eq@decide@bool{#3}{#2}{#1}\eq@equations@end@modifier
5572 \eq@decide@bool{#3}{#2}{#1}\eq@box@end@modifier}
5573 \eq@define@key{setup}{modifiereqncr}[true]{\eq@decide@if{#3}{#2}{#1}%
5574 {\eq@multi@cr@test@setall\eq@break@cr@test@setall}
5575 \eq@box@cr@test@setall}%
5576 {\eq@multi@cr@test@setopt\eq@break@cr@test@setopt
5577 \eq@box@cr@test@setopt}}
5578 \eq@define@key{setup}{modifiereqnamp}[true]{\eq@decide@if{#3}{#2}{#1}%
5579 {\eq@columns@amp@test@setall\eq@break@amp@test@setall
5580 \eq@box@amp@test@setall}%
5581 {\eq@columns@amp@test@setopt\eq@break@amp@test@setopt
5582 \eq@box@amp@test@setopt}}
5583 \eq@define@key{setup}{modifierbreak}[true]{\eq@decide@if{#3}{#2}{#1}%
5584 \eq@break@test@setall\eq@break@test@setopt}
5585 \eq@define@key{setup}{modifierwarning}[all]{%
5586 \eq@decide@select{#3}{#2}{#1}{%
5587 {\eq@decide@false{\let\eq@parseopt@warn@env\@empty
5588 \let\eq@parseopt@warn@cr\@empty}},%
5589 {\env,-}{\let\eq@parseopt@warn@env\eq@warn@parseopt
5590 \let\eq@parseopt@warn@cr\@empty}},%
5591 {all,\eq@decide@true}{\let\eq@parseopt@warn@env\eq@warn@parseopt
5592 \let\eq@parseopt@warn@cr\eq@warn@parseopt}},%
5593 {verbose,+}{\let\eq@parseopt@warn@env\eq@warn@parseopt@verbose
5594 \let\eq@parseopt@warn@cr\eq@warn@parseopt@verbose}}}}

5595 \eq@define@key{setup}{modifiereqn}[true]{%
5596 \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier
5597 \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier
5598 \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier}
5599 \eq@define@key{setup}{modifiereqnenv}[true]{%
5600 \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier}
5601 \eq@define@key{setup}{modifiereqnsqr}[true]{%
5602 \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier}
5603 \eq@define@key{setup}{modifiereqnang}[true]{%
5604 \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier}
5605 \eq@define@key{setup}{modifiereqnend}[true]{%
5606 \eq@decide@bool{#3}{#2}{#1}\eq@equations@end@modifier}
5607 \eq@define@key{setup}{modifiereqncr}[true]{\eq@decide@if{#3}{#2}{#1}%
5608 {\eq@multi@cr@test@setall\eq@break@cr@test@setall}%
5609 {\eq@multi@cr@test@setopt\eq@break@cr@test@setopt}}
5610 \eq@define@key{setup}{modifiereqnamp}[true]{\eq@decide@if{#3}{#2}{#1}%
5611 {\eq@columns@amp@test@setall\eq@break@amp@test@setall}%
5612 {\eq@columns@amp@test@setopt\eq@break@amp@test@setopt}}

5613 \eq@define@key{setup}{modifierbox}[true]{%
5614 \eq@decide@bool{#3}{#2}{#1}\eq@box@env@modifier

```

```

5615 \eql@decide@bool{#3}{#2}{#1}\eql@box@ang@modifier}
5616 \eql@define@key{setup}{modifierboxenv}[true]{%
5617 \eql@decide@bool{#3}{#2}{#1}\eql@box@env@modifier}
5618 \eql@define@key{setup}{modifierboxang}[true]{%
5619 \eql@decide@bool{#3}{#2}{#1}\eql@box@ang@modifier}
5620 \eql@define@key{setup}{modifierboxend}[true]{%
5621 \eql@decide@bool{#3}{#2}{#1}\eql@box@end@modifier}
5622 \eql@define@key{setup}{modifierboxcr}[true]{%
5623 \eql@decide@if{#3}{#2}{#1}\eql@box@cr@test@setall\eql@box@cr@test@setopt}
5624 \eql@define@key{setup}{modifierboxamp}[true]{%
5625 \eql@decide@if{#3}{#2}{#1}\eql@box@amp@test@setall\eql@box@amp@test@setopt}

5626 \eql@define@key{equations}{modifierend}[true]{%
5627 \eql@decide@bool{#3}{#2}{#1}\eql@equations@end@modifier}
5628 \eql@define@key{equations}{modifiercr}[true]{\eql@decide@if{#3}{#2}{#1}%
5629 {\eql@multi@cr@test@setall\eql@break@cr@test@setall}%
5630 {\eql@multi@cr@test@setopt\eql@break@cr@test@setopt}}
5631 \eql@define@key{equations}{modifieramp}[true]{\eql@decide@if{#3}{#2}{#1}%
5632 {\eql@columns@amp@test@setall\eql@break@amp@test@setall}%
5633 {\eql@columns@amp@test@setopt\eql@break@amp@test@setopt}}
5634 \eql@define@key{setup,equations}{crrelnext}[true]{%
5635 \eql@decide@bool{#3}{#2}{#1}\eql@multi@cr@relnext}

5636 \eql@define@key{box}{modifier}[true]{\eql@decide@if{#3}{#2}{#1}%
5637 {\eql@box@cr@test@setall\eql@box@amp@test@setall
5638 \let\eql@box@end@modifier\eql@true}%
5639 {\eql@box@cr@test@setopt\eql@box@amp@test@setopt
5640 \let\eql@box@end@modifier\eql@false}}
5641 \eql@define@key{box}{modifierend}[true]{%
5642 \eql@decide@bool{#3}{#2}{#1}\eql@box@end@modifier}
5643 \eql@define@key{box}{modifiercr}[true]{%
5644 \eql@decide@if{#3}{#2}{#1}\eql@box@cr@test@setall\eql@box@cr@test@setopt}
5645 \eql@define@key{box}{modifieramp}[true]{%
5646 \eql@decide@if{#3}{#2}{#1}\eql@box@amp@test@setall\eql@box@amp@test@setopt}

```

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

```

5647 \eql@define@key\eql@keyall{spread}{%
5648 \let\eql@spread@reset\eql@false\def\eql@spread@val{#1}}
5649 \eql@define@key\eql@keyall{spread*}[Opt]{%
5650 \let\eql@spread@reset\eql@true\def\eql@spread@val{#1}}
5651 \eql@define@key\eql@keyall{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
5652 {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
5653 {{cell}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
5654 {{tag}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
5655 {\eql@decide@true
5656 {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
5657 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

5658 \eql@define@key\eql@keyall{displayheight}[strut]{%
5659 \eql@decide@select{#3}{#2}{#1}{%
5660 {\eql@decide@false{\let\eql@display@height@undefined}},%
5661 {{strut}{\def\eql@display@height{\ht\eql@strutbox@}}},%
5662 {\relax{\def\eql@display@height{#1}}}}
5663 \eql@define@key\eql@keyall{displaydepth}[strut]{%
5664 \eql@decide@select{#3}{#2}{#1}{%

```

```

5665 \eql@decide@false{\let\eqldisplay@depth\undefined}},%
5666 {{strut}{\def\eqldisplay@depth{\dp\eql@strutbox@}}},%
5667 {\relax{\def\eqldisplay@depth{#1}}}}

```

Settings concerning page breaks:

```

5668 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5669   {{force,4,\eql@decide@true}{\eqldisplaybreak@pre4}},%
5670   {{high,3}{\eqldisplaybreak@pre3}},%
5671   {{med,medium,2}{\eqldisplaybreak@pre2}},%
5672   {{low,1}{\eqldisplaybreak@pre1}},%
5673   {{0,\eql@decide@false}{\eqldisplaybreak@pre0}},%
5674   {{default,inherit,-1}{\eqldisplaybreak@pre@m@ne}}}}
5675 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5676   {{force,4,\eql@decide@true}{\eqldisplaybreak@post4}},%
5677   {{high,3}{\eqldisplaybreak@post3}},%
5678   {{med,medium,2}{\eqldisplaybreak@post2}},%
5679   {{low,1}{\eqldisplaybreak@post1}},%
5680   {{0,\eql@decide@false}{\eqldisplaybreak@post0}},%
5681   {{default,inherit,-1}{\eqldisplaybreak@post@m@ne}}}}
5682 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
5683   \eql@decide@select{#3}{#2}{#1}{%
5684     {{full,4}{\eqldisplaybreak@inter4}},%
5685     {{high,3}{\eqldisplaybreak@inter3}},%
5686     {{med,medium,2}{\eqldisplaybreak@inter2}},%
5687     {{low,1}{\eqldisplaybreak@inter1}},%
5688     {{0,\eql@decide@false}{\eqldisplaybreak@inter\z@}}}}
5689 \eql@define@key{equations}{prepenalty}{%
5690   \eqldisplaybreak@prepen@numexpr#1\relax}
5691 \eql@define@key{equations}{postpenalty}{%
5692   \eqldisplaybreak@postpen@numexpr#1\relax}
5693 \eql@define@key{equations,setup}{interpenalty}{%
5694   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

5695 \eql@define@key{control}{vspace}[]{\eql@vspace@add{#1}}
5696 \eql@define@key{control}{vspace*}[]{\eql@vspace@addfixedbefore{#1}}
5697 \eql@define@key{control}{vspace!}[]{\eql@vspace@addfixedafter{#1}}
5698 \eql@define@key{control}{break}[4]{\eqldisplaybreak@level[#{#1}]}
5699 \eql@define@key{control}{penalty}[]{\eqldisplaybreak@star{#1}}

```

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

```

5700 \eql@define@key{equations}{noskip}[both]{%
5701   \eql@decide@abovebelow{#3}{#2}{#1}%
5702   {\def\eql@skip@force@above{5}}%
5703   {\def\eql@skip@force@below{5}}}}
5704 \eql@define@key{equations}{short}[above]{%
5705   \eql@decide@abovebelow{#3}{#2}{#1}%
5706   {\def\eql@skip@force@above{1}}%
5707   {\def\eql@skip@force@below{1}}}}
5708 \eql@define@key{equations}{long}[both]{%
5709   \eql@decide@abovebelow{#3}{#2}{#1}%
5710   {\def\eql@skip@force@above{0}}%
5711   {\def\eql@skip@force@below{0}}}}
5712 \eql@define@key{equations}{medskip}[both]{%
5713   \eql@decide@abovebelow{#3}{#2}{#1}%
5714   {\def\eql@skip@force@above{6}}%

```

```

5715   {\def\eql@skip@force@below{6}}
5716 \eql@define@key{equations}{par}[par]{%
5717   \eql@decide@select{#3}{#2}{#1}{%
5718     {{default,\eql@decide@false}}{\let\eql@skip@force@leave\@undefined}},%
5719     {{cont,hmode}}{\let\eql@skip@force@leave\z@}},%
5720     {{par,vmode}}{\let\eql@skip@force@leave\@one
5721       \ifdefined\eql@skip@force@below\else
5722         \def\eql@skip@force@below{3}%
5723         \fi}},%
5724     {{top}}{\let\eql@skip@force@leave\tw@
5725       \ifdefined\eql@skip@force@below\else
5726         \def\eql@skip@force@below{4}%
5727         \fi}}}}

```

Specify vertical spacing explicitly:

```

5728 \eql@define@key{equations}{skip}{%
5729   \def\eql@skip@force@above{7}%
5730   \def\eql@skip@custom@above{#1}%
5731   \let\eql@skip@force@below\eql@skip@force@above
5732   \let\eql@skip@custom@below\eql@skip@custom@above}
5733 \eql@define@key{equations}{aboveskip}{%
5734   \def\eql@skip@force@above{7}%
5735   \def\eql@skip@custom@above{#1}}
5736 \eql@define@key{equations}{belowskip}{%
5737   \def\eql@skip@force@below{7}%
5738   \def\eql@skip@custom@below{#1}}
5739 \eql@define@key{equations}{abovespace}{%
5740   \advance\eql@abovespace@\glueexpr#1\relax}
5741 \eql@define@key{equations}{belowspace}{%
5742   \advance\eql@belowspace@\glueexpr#1\relax}

```

Vertical spacing for intertext:

```

5743 \eql@define@key{intertext}{skip}{%
5744   \def\eql@skip@force@above{7}%
5745   \def\eql@skip@custom@above{#1}%
5746   \let\eql@skip@force@below\eql@skip@force@above
5747   \let\eql@skip@custom@below\eql@skip@custom@above}
5748 \eql@define@key{intertext}{aboveskip}{%
5749   \def\eql@skip@force@below{7}%
5750   \def\eql@skip@custom@below{#1}}
5751 \eql@define@key{intertext}{belowskip}{%
5752   \def\eql@skip@force@above{7}%
5753   \def\eql@skip@custom@above{#1}}
5754 \eql@define@key{intertext}{noskip}[both]{%
5755   \eql@decide@abovebelow{#3}{#2}{#1}%
5756   {\def\eql@skip@force@below{5}}%
5757   {\def\eql@skip@force@above{5}}}
5758 \eql@define@key{intertext}{short}[both]{%
5759   \eql@decide@abovebelow{#3}{#2}{#1}%
5760   {\def\eql@skip@force@below{1}}%
5761   {\def\eql@skip@force@above{1}}}
5762 \eql@define@key{intertext}{long}[both]{%
5763   \eql@decide@abovebelow{#3}{#2}{#1}%
5764   {\def\eql@skip@force@below{0}}%
5765   {\def\eql@skip@force@above{0}}}
5766 \eql@define@key{intertext}{medskip}[both]{%
5767   \eql@decide@abovebelow{#3}{#2}{#1}%
5768   {\def\eql@skip@force@below{6}}%

```

5769 {\def\eq@skip@force@above{6}}

Configure general vertical spacing behaviour for various situations:

```
5770 \eq@define@key{setup}{skip, longskip}{%
5771   \abovedisplayskip\glueexpr#1\relax
5772   \belowdisplayskip\abovedisplayskip
5773   \def\eq@skip@long@above{#1}%
5774   \let\eq@skip@long@below\eq@skip@long@above}
5775 \eq@define@key{setup}{aboveskip, abovelongskip}{%
5776   \abovedisplayskip\glueexpr#1\relax
5777   \def\eq@skip@long@above{#1}}
5778 \eq@define@key{setup}{belowskip, belowlongskip}{%
5779   \belowdisplayskip\glueexpr#1\relax
5780   \def\eq@skip@long@below{#1}}
5781 \eq@define@key{setup}{aboveshortskip}{%
5782   \abovedisplayshortskip\glueexpr#1\relax
5783   \def\eq@skip@short@above{#1}}
5784 \eq@define@key{setup}{belowshortskip}{%
5785   \belowdisplayshortskip\glueexpr#1\relax
5786   \def\eq@skip@short@below{#1}}
5787 \eq@define@key{setup}{tagskip}{%
5788   \def\eq@skip@tag@above{#1}%
5789   \let\eq@skip@tag@below\eq@skip@tag@above}
5790 \eq@define@key{setup}{abovetagskip}{%
5791   \def\eq@skip@tag@above{#1}}
5792 \eq@define@key{setup}{belowtagskip}{%
5793   \def\eq@skip@tag@below{#1}}
5794 \eq@define@key{setup}{medskip}{%
5795   \def\eq@skip@med@above{#1}%
5796   \let\eq@skip@med@below\eq@skip@med@above}
5797 \eq@define@key{setup}{abovemedskip}{%
5798   \def\eq@skip@med@above{#1}}
5799 \eq@define@key{setup}{belowmedskip}{%
5800   \def\eq@skip@med@below{#1}}
5801 \eq@define@key{setup}{abovetopskip}{%
5802   \def\eq@skip@top@above{#1}}
5803 \eq@define@key{setup}{belowtopskip}{%
5804   \def\eq@skip@top@below{#1}}
5805 \eq@define@key{setup}{aboveparskip}{%
5806   \def\eq@skip@par@above{#1}}
5807 \eq@define@key{setup}{belowparskip}{%
5808   \def\eq@skip@par@below{#1}}
5809 \eq@define@key{setup}{abovecontskip}{%
5810   \eq@decide@select{#3}{#2}{#1}{%
5811     {\hide}{\def\eq@skip@cont@above{\eq@spread@val-\eq@skip@long@below}}},%
5812     {\relax{\def\eq@skip@cont@above{#1}}}}}}
5813 \eq@define@key{setup}{belowcontskip}{%
5814   \def\eq@skip@cont@below{#1}}
5815 \eq@define@key{setup}{shortmode}{%
5816   \eq@decide@select{#3}{#2}{#1}{%
5817     {\eq@decide@false, never}{\def\eq@skip@mode@short{0}}},%
5818     {\above, neverbelow, belowoff}{\def\eq@skip@mode@short{1}}},%
5819     {\belowone, belowsingle}{\def\eq@skip@mode@short{2}}},%
5820     {\belowall, always, on}{\def\eq@skip@mode@short{3}}}}}}
5821 \eq@define@key{setup}{abovecontmode}{%
5822   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
5823 \eq@define@key{setup}{belowcontmode}{%
5824   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
```

```

5825 \eql@define@key{setup}{aboveparmode}{%
5826   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
5827 \eql@define@key{setup}{belowparmode}{%
5828   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
5829 \eql@define@key{setup}{abovetopmode}{%
5830   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}
5831 \eql@define@key{setup}{belowtopmode}{%
5832   \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@below}

```

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

```

5833 \eql@define@key{equations,subequations}{label}{\eql@tags@addblock@label{#1}}
5834 \eql@define@key{equations,subequations}{labelname}{\eql@tags@addblock@name{#1}}
5835 \eql@define@key{equations,subequations}{tag}{\eql@tags@addblock@tag{#1}}
5836 \eql@define@key{equations,subequations}{tag*}{%
5837   \eql@tags@addblock@tagform@off\eql@tags@addblock@tag{#1}}
5838 \eql@define@key{equations,subequations}{taglabel}{\eql@tags@addblock@ref{#1}}

```

TODO: describe

```

5839 \eql@define@key{control}{label}{\eql@tags@add@label{#1}}
5840 \eql@define@key{control}{labelname}{\eql@tags@add@name{#1}}
5841 \eql@define@key{control}{tag}{\eql@tags@add@tag{#1}}
5842 \eql@define@key{control}{tag*}{\eql@tags@add@tagform@off\eql@tags@add@tag{#1}}
5843 \eql@define@key{control}{taglabel}{\eql@tags@add@ref{#1}}
5844 \eql@define@key{control}{shifftag}{\eql@tags@add@raiseshift{#1}}
5845 \eql@define@key{control}{smashtag}{\eql@tags@add@raisesmash{#1}}
5846 \eql@define@key{control}{pushtag}{\eql@tags@add@forceraise}

```

TODO: describe

```

5847 \eql@define@key{setup}{labelname}{\protected@edef\eql@tags@name@generic{#1}}
5848 \eql@define@key{setup}{autolabel}[true]{%
5849   \eql@decide@bool{#3}{#2}{#1}\eql@tags@autolabel}
5850 \eql@define@key{setup}{autotag}[true]{%
5851   \eql@decide@bool{#3}{#2}{#1}\eql@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

5852 \eql@define@key{equations,setup}{tagmargin}[auto]{%
5853   \eql@decide@select{#3}{#2}{#1}{%
5854     {auto,\eql@decide@false}{\let\eql@tagmargin@val\undefined}},%
5855     {\relax{\def\eql@tagmargin@val{#1}}}}
5856 \eql@define@key{equations,setup}{tagmargin*}{%
5857   \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}}
5858 \eql@define@key{equations,setup}{tagmarginratio}{%
5859   \eql@tagmargin@ratio@{dimexpr#1pt\relax}}
5860 \eql@define@key{equations,setup}{tagmarginthreshold}{%
5861   \def\eql@tagmargin@threshold{#1}}
5862 \eql@define@key{equations,setup}{mintagsep}{\def\eql@tagsepmin@val{#1}}
5863 \eql@define@key{equations,setup}{mintagwidth}{%
5864   \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}}
5865 \eql@define@key{equations,setup}{mintagwidth*}{%
5866   \settowidth\eql@tagwidthmin@{#1}}
5867 \eql@define@key{equations,setup}{tagsnap}{%
5868   \eql@decide@select{#3}{#2}{#1}{%
5869     {\eql@decide@false}{\let\eql@tagpos@snap\z@}},%
5870     {\relax{\def\eql@tagpos@snap{#1}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

5871 \eql@define@key{equations,setup}{tagbox,taglayout}{%
5872   \eql@tags@taglayout@set{#1}}
5873 \eql@define@key{equations,setup}{tagbox*,taglayout*}{%
5874   \eql@tags@taglayout@set@direct{#1}}
5875 \eql@define@key{equations,setup}{tagform}{%
5876   \eql@tags@tagform@set#1}
5877 \eql@define@key{equations,setup}{tagform*}{%
5878   \eql@tags@tagform@set@direct{#1}}
5879 \eql@define@key{equations,setup}{subeqtemplate}{%
5880   \eql@subequations@template@set{#1}}

5881 \eql@define@key{control}{tagbox,taglayout}{%
5882   \global\eql@append\eql@tags@container{\eql@tags@taglayout@set{#1}}}
5883 \eql@define@key{control}{tagbox*,taglayout*}{%
5884   \global\eql@append\eql@tags@container{\eql@tags@taglayout@set@direct{#1}}}
5885 \eql@define@key{control}{tagform}{%
5886   \global\eql@append\eql@tags@container{\eql@tags@tagform@set#1}}
5887 \eql@define@key{control}{tagform*}[{###1}]{%
5888   \global\eql@append\eql@tags@container{\eql@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

5889 \eql@define@key{equations,setup}{numberline,number,num,numline,n}[all]{%
5890   \eql@decide@select{#3}{#2}{#1}{%
5891     {\eql@decide@false,0,*}{\let\eql@numbering@active\eql@false}},%
5892     {\eql@decide@true,!}{\let\eql@numbering@active\eql@true}},%
5893     {none,n,-}{\let\eql@numbering@mode\eql@numbering@mode@multi
5894       \let\eql@numbering@active\eql@false}},%
5895     {single,1}{\let\eql@numbering@mode\eql@numbering@mode@single
5896       \let\eql@numbering@active\eql@true}},%
5897     {multi,@}{\let\eql@numbering@mode\eql@numbering@mode@multi
5898       \let\eql@numbering@active\eql@true}},%
5899     {\relax{\eql@numbering@set{#1}}}}
5900 \eql@define@key{equations,setup}{nonumber,nn,*}[]{%
5901   \let\eql@numbering@active\eql@false}
5902 \eql@define@key{equations,setup}{donumber,dn,!}[]{%
5903   \let\eql@numbering@active\eql@true}
5904 \eql@define@key{equations,setup}{tagsleft,leqno}[]{%
5905   \let\eql@tagsleft\eql@true}
5906 \eql@define@key{equations,setup}{tagsright,reqno}[]{%
5907   \let\eql@tagsleft\eql@false}
5908 \eql@define@key{equations,setup}{tags,eqno}{%
5909   \eql@decide@select{#3}{#2}{#1}{%
5910     {right,r}{\let\eql@tagsleft\eql@false}},%
5911     {left,l}{\let\eql@tagsleft\eql@true}}}}
5912 \eql@define@key{equations,setup}{evadetag,avoidtag}[true]{%
5913   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}
5914 \eql@define@key{equations,setup}{tagbetween}[true]{%
5915   \eql@decide@bool{#3}{#2}{#1}\eql@tagpos@doconvert}

```

TODO: describe

```

5916 \eql@define@key{control}{nonumber,nn,*}[]{\global\@eqnswfalse}
5917 \eql@define@key{control}{donumber,dn,!}[]{\global\@eqnswtrue}
5918 \eql@define@key{control}{numberhere}[]{\eql@numberhere}
5919 \eql@define@key{control}{numbernext}[]{\eql@numbernext}

```

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

```

5920 \eql@define@key{equations,setup}{layout}{\eql@decide@select{#3}{#2}{#1}{%
5921   {{center,c}{\let\eql@layoutleft\eql@false}},%
5922   {{left,l}{\let\eql@layoutleft\eql@true}}}}
5923 \eql@define@key{equations,setup}{center}[]{%
5924   \let\eql@layoutleft\eql@false}
5925 \eql@define@key{equations,setup}{flushleft,left}[]{%
5926   \let\eql@layoutleft\eql@true}
5927 \eql@define@key{equations,setup}{leftmargin}{\def\eql@layoutleftmargin{#1}}
5928 \eql@define@key{equations,setup}{leftmargin*}{%
5929   \settoheight\dimen{#1}\edef\eql@layoutleftmargin{\the\dimen}}
5930 \eql@define@key{equations,setup}{minleftmargin}{%
5931   \def\eql@layoutleftmarginmin{#1}}
5932 \eql@define@key{equations,setup}{maxleftmargin}{%
5933   \eql@decide@select{#3}{#2}{#1}{%
5934     {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
5935     {\relax{\def\eql@layoutleftmarginmax{#1}}}}}}

5936 \eql@define@key{equations,box}{margin}{%
5937   \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
5938 \eql@define@key{equations,box}{marginleft}{\def\eql@display@marginleft{#1}}
5939 \eql@define@key{equations,box}{marginright}{\def\eql@display@marginright{#1}}
5940 \eql@define@key{equations}{linewidth,width}{\def\eql@display@linewidth{#1}}

```

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

```

5941 \eql@define@key{equations,setup}{alignshrink}{\eql@decide@select{#3}{#2}{#1}{%
5942   {{max,full,4}{\eql@alignbadness@inf@bad}},%
5943   {{high,3}{\eql@alignbadness@54\relax}},%
5944   {{med,medium,2}{\eql@alignbadness@18\relax}},%
5945   {{low,1}{\eql@alignbadness@6\relax}},%
5946   {{0,\eql@decide@false}{\eql@alignbadness@z@}}}}
5947 \eql@define@key{equations,setup}{tagshrink}{\eql@decide@select{#3}{#2}{#1}{%
5948   {{max,full,4}{\eql@tagbadness@inf@bad}},%
5949   {{high,3}{\eql@tagbadness@54\relax}},%
5950   {{med,medium,2}{\eql@tagbadness@18\relax}},%
5951   {{low,1}{\eql@tagbadness@6\relax}},%
5952   {{0,\eql@decide@false}{\eql@tagbadness@z@}}}}
5953 \eql@define@key{equations,setup}{alignbadness}{%
5954   \eql@alignbadness@numexpr#1\relax}
5955 \eql@define@key{equations,setup}{tagbadness}{%
5956   \eql@tagbadness@numexpr#1\relax}
5957 \eql@define@key{equations,setup}{mincolsep}{\eql@decide@select{#3}{#2}{#1}{%
5958   {{0,\eql@decide@false}{\def\eql@colsepmin@val{Opt}}},%
5959   {\relax{\def\eql@colsepmin@val{#1}}}}}}
5960 \eql@define@key{equations,setup}{maxcolsep}{\eql@decide@select{#3}{#2}{#1}{%
5961   {\eql@decide@false{\def\eql@colsepmax@val{.5\maxdimen}}},%
5962   {0{\def\eql@colsepmax@val{Opt}}},%
5963   {\relax{\def\eql@colsepmax@val{#1}}}}}}
5964 \eql@define@key{equations,setup}{fulllength}[true]{%
5965   \eql@decide@bool{#3}{#2}{#1}\eql@columns@fulllength}

```

TODO: is boxcolsep vs breakcolsep okay??!

```

5966 \eql@define@key{equations,setup}{linesep}{\eql@decide@select{#3}{#2}{#1}{%
5967   {{0,\eql@decide@false}{\def\eql@break@line@sep{Opt}}},%

```

```

5968   {\relax{\def\eq@break@line@sep{#1}}}}
5969 \eq@define@key{equations,setup}{linesep*}{\eq@decide@select{#3}{#2}{#1}{%
5970   {0,\eq@decide@false}{\def\eq@break@line@shortsep{Opt}}},%
5971   {\relax{\def\eq@break@line@shortsep{#1}}}}
5972 \eq@define@key{box,setup}{colsep}{\eq@decide@select{#3}{#2}{#1}{%
5973   {0,\eq@decide@false}{\def\eq@box@colsep{Opt}}},%
5974   {short,*}{\def\eq@box@colsep{\eq@box@shortsep}}},%
5975   {\relax{\def\eq@box@colsep{#1}}}}%
5976   \let\eq@break@col@sep\eq@box@colsep}
5977 \eq@define@key{equations}{colsep}{\eq@decide@select{#3}{#2}{#1}{%
5978   {0,\eq@decide@false}{\def\eq@break@col@sep{Opt}}},%
5979   {\relax{\def\eq@break@col@sep{#1}}}}%
5980   \let\eq@colsepmin@val\eq@box@colsep
5981   \let\eq@colsepmax@val\eq@box@colsep
5982   \let\eq@box@colsep\eq@break@col@sep}
5983 \eq@define@key{equations,setup}{colsep*}{\eq@decide@select{#3}{#2}{#1}{%
5984   {0,\eq@decide@false}{\def\eq@break@col@shortsep{Opt}}},%
5985   {\relax{\def\eq@break@col@shortsep{#1}}}}}}
5986 \eq@define@key{box,setup}{colsep*}{\eq@decide@select{#3}{#2}{#1}{%
5987   {0,\eq@decide@false}{\def\eq@box@shortsep{Opt}}},%
5988   {\relax{\def\eq@box@shortsep{#1}}}}}}
5989 \eq@define@key{box,setup}{condsep}{\eq@decide@select{#3}{#2}{#1}{%
5990   {0,\eq@decide@false}{\def\eq@box@condsep{Opt}}},%
5991   {short,*}{\def\eq@box@condsep{\eq@box@shortsep}}},%
5992   {\relax{\def\eq@box@condsep{#1}}}}}}
5993 \eq@define@key{box,setup}{matrixsep}{\eq@decide@select{#3}{#2}{#1}{%
5994   {0,\eq@decide@false}{\def\eq@box@matrixsep{Opt}}},%
5995   {short,*}{\def\eq@box@matrixsep{\eq@box@shortsep}}},%
5996   {\relax{\def\eq@box@matrixsep{#1}}}}}}

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5997 \eq@define@key\eq@keyall{shape}[default]{\eq@shape@set{#1}}
5998 \eq@define@key\eq@keyall{padding,pad}[indent]{%
5999   \eq@decide@select{#3}{#2}{#1}{%
6000     {max}{\let\eq@paddingleft@val\@undefined}},%
6001     {indent}{\def\eq@paddingleft@val{\eq@indent@val}}},%
6002     {0,\eq@decide@false}{\def\eq@paddingleft@val{Opt}}},%
6003     {\relax{\def\eq@paddingleft@val{#1}}}}%
6004   \let\eq@paddingright@val\eq@paddingleft@val}
6005 \eq@define@key\eq@keyall{padleft}[indent]{%
6006   \eq@decide@select{#3}{#2}{#1}{%
6007     {max}{\let\eq@paddingleft@val\@undefined}},%
6008     {indent}{\def\eq@paddingleft@val{\eq@indent@val}}},%
6009     {0,\eq@decide@false}{\def\eq@paddingleft@val{Opt}}},%
6010     {\relax{\def\eq@paddingleft@val{#1}}}}}}
6011 \eq@define@key\eq@keyall{padright}[indent]{%
6012   \eq@decide@select{#3}{#2}{#1}{%
6013     {max}{\let\eq@paddingright@val\@undefined}},%
6014     {indent}{\def\eq@paddingright@val{\eq@indent@val}}},%
6015     {0,\eq@decide@false}{\def\eq@paddingright@val{Opt}}},%
6016     {\relax{\def\eq@paddingright@val{#1}}}}}}
6017 \eq@define@key\eq@keyall{indent}[2em]{%
6018   \def\eq@indent@val{#1}}

```

TODO: describe

```

6019 \def\eq@shape@rel{\eqncontrol{align=left}}
6020 \def\eq@shape@cont{\eqncontrol{align=left,shift=*}}

```

```

6021 \eql@define@key\eql@keyall{shaperel}[]{\def\eql@shape@rel{#1}}
6022 \eql@define@key\eql@keyall{shapecont}[]{\def\eql@shape@cont{#1}}

```

TODO: describe

```

6023 \eql@define@key{control}{align}[]{%
6024   \eql@decide@select{#3}{#2}{#1}{%
6025     {l,left}{\global\eql@append\eql@cell@container{\eql@shape@pos@z}}},%
6026     {c,center}{\global\eql@append\eql@cell@container{\eql@shape@pos@ne}}},%
6027     {r,right}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw}}}}%
6028 \eql@define@key{control}{shift,shifto}[]{%
6029   \eql@decide@select{#3}{#2}{#1}{%
6030     {*,indent}{\eql@shape@alignamount@set{\eql@indent}}},%
6031     {!,outdent}{\eql@shape@alignamount@set{-\eql@indent}}},%
6032     {relax{\eql@shape@alignamount@set{#1}}}}%
6033 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

6034 \eql@define@key\eql@keyall{classout}{\eql@class@innerleft@set{#1}}
6035 \eql@define@key\eql@keyall{classin}{\eql@class@innerright@set{#1}}
6036 \eql@define@key\eql@keyall{classlead,classin*}{\eql@class@innerlead@set{#1}}
6037 \eql@define@key\eql@keyall{rel}{\def\eql@class@rel@symb{#1}}
6038 \eql@define@key\eql@keyall{classbreak}{\eql@class@rel@break@set{#1}}
6039 \eql@define@key\eql@keyall{classtart}{\eql@class@rel@start@set{#1}}
6040 \eql@define@key\eql@keyall{classbreakskip}[adjust]{%
6041   \eql@decide@select{#3}{#2}{#1}{%
6042     {0,\eql@decide@false}{\def\eql@class@rel@break@skip{0mu}}},%
6043     {adjust,*}{\def\eql@class@rel@break@skip{\thickmuskip-\medmuskip}}},%
6044     {relax{\def\eql@class@rel@break@skip{#1}}}}%
6045 \eql@define@key{control}{rel}[\eql@class@rel@symb]{\eql@class@rel@make{#1}}
6046 \eql@define@key{control}{rel;}[]{\eql@class@rel@make{}}
6047 \eql@define@key{control}{rel*}[]{\eql@class@rel@make{}}
6048 \eql@define@key\eql@keyall{ampeq}[]{\eql@class@ampeq}
6049 \eql@define@key\eql@keyall{eqamp}[]{\eql@class@eqamp}
6050 \eql@define@key\eql@keyall{class}{\eql@decide@select{#3}{#2}{#1}{%
6051   {ampeq,amprel,eqafter,beforerel}\eql@class@ampeq},%
6052   {eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}%

```

Math Styles. Configure math classes at alignment marker:

```

6053 \eql@define@key\eql@keyall{style}[display]{%
6054   \eql@decide@select{#3}{#2}{#1}{%
6055     {text}{\let\eql@mathstyle\@empty}}},%
6056     {display}{\let\eql@mathstyle\displaystyle}}%
6057 \eql@define@key{setup}{casesstyle}[display]{%
6058   \eql@decide@select{#3}{#2}{#1}{%
6059     {\eql@decide@false{\let\eql@cases@mathstyle\eql@false}}},%
6060     {text}{\let\eql@cases@mathstyle\@empty}}},%
6061     {display}{\let\eql@cases@mathstyle\displaystyle}}%

```

Punctuation. Configure punctuation defaults: **TODO:** describe

```

6062 \def\eql@punct@all#1#2#3#4#5\eql@punct@end{%
6063   \def\eql@tmp{#4}\def\eql@tmpa{1}%
6064   \ifx\eql@tmp\eql@tmpa
6065     \ifnum#5=1111\relax
6066     \eql@punct@set\eql@punct@col{#1}%

```

```

6067 \eql@punct@set\eql@punct@line{#2}%
6068 \eql@punct@set\eql@punct@block{#3}%
6069 \else\ifnum#5=111\relax
6070 \eql@punct@set\eql@punct@line{#1}%
6071 \eql@punct@set\eql@punct@block{#2}%
6072 \else\ifnum#5=11\relax
6073 \eql@punct@set\eql@punct@block{#1}%
6074 \else
6075 \eql@punct@clear
6076 \fi\fi\fi
6077 \else
6078 \eql@error{Too many arguments to punctall}%
6079 \fi
6080 }

```

TODO: describe

```

6081 \eql@define@key\eql@keyall{punctsep}[\,]{\def\eql@punct@sep{#1}}
6082 \eql@define@key\eql@keyall{punctclass}[\mathclose{}]{\def\eql@punct@class{#1}}
6083 \eql@define@key\eql@keyall{punct}[.]{\eql@punct@set\eql@punct@block{#1}}
6084 \eql@define@key\eql@keyall{punct*}[]{\eql@punct@set\eql@punct@block\relax}
6085 \eql@define@key\eql@keyall{punctline}[,]{\eql@punct@set\eql@punct@line{#1}}
6086 \eql@define@key\eql@keyall{punctline*}[]{\eql@punct@set\eql@punct@line\relax}
6087 \eql@define@key\eql@keyall{punctcol}[,]{\eql@punct@set\eql@punct@col{#1}}
6088 \eql@define@key\eql@keyall{punctcol*}[]{\eql@punct@set\eql@punct@col\relax}
6089 \eql@define@key{box}{punctcases}[,]{\eql@punct@set\eql@punct@col{#1}}
6090 \eql@define@key{box}{punctcases*}[]{\eql@punct@set\eql@punct@col\relax}
6091 \eql@define@key{setup}{punctcases}[,]{\eql@punct@set\eql@punct@cases{#1}}
6092 \eql@define@key{setup}{punctcases*}[]{\eql@punct@set\eql@punct@cases\relax}
6093 \eql@define@key\eql@keyall{punctall}[,,]{\eql@punct@all#111111\eql@punct@end}
6094 \eql@define@key{box}{punctterm}[true]{%
6095 \eql@decide@bool{#3}{#2}{#1}\eql@box@punct@term}

6096 \eql@define@key{control}{punctsep}[\,]{\def\eql@punct@sep{#1}}
6097 \eql@define@key{control}{setpunct}[.]{\eql@punct@set\eql@punct@next{#1}}
6098 \eql@define@key{control}{setpunct}[.]{\eql@punct@set\eql@punct@next{#1}}
6099 \eql@define@key{control}{setpunct*}[]{\let\eql@punct@next\relax}
6100 \eql@define@key{control}{punct,punctapply}[\relax]{%
6101 \eql@punct@set\eql@punct@next{#1}\eql@punct@apply@top}
6102 \eql@define@key{control}{punctline}[]{\eql@punct@print@line}
6103 \eql@define@key{control}{punctcol}[]{\eql@punct@print@col}

```

Frames. **TODO:** describe

```

6104 \eql@define@key{box}{frame}[\fbox]{%
6105 \def\eql@box@frame{#1}%
6106 \ifx\eql@box@frame@empty\let\eql@box@frame@firstofone\fi}
6107 \eql@define@key{box}{wrap}{\eql@box@wrap#1}
6108 \eql@define@key{box}{delim}[r]{\eql@decide@delim{#3}{#2}{#1}}
6109 \eql@define@key{box}{ldelim}{\eql@box@ldelim#1}
6110 \eql@define@key{box}{rdelim}{\eql@box@rdelim#1}
6111 \eql@define@key{box}{lbrace}[]{\eql@box@ldelim\lbrace}
6112 \eql@define@key{box}{rbrace}[]{\eql@box@rdelim\rbrace}
6113 \eql@define@key{box}{lbrace,lrbaces}[]{\eql@box@delim\lbrace\rbrace}
6114 \eql@define@key{box}{braces}[lr]{%
6115 \eql@decide@select{#3}{#2}{#1}{%
6116 {\eql@decide@false}{\eql@box@wrap}{}}},%
6117 {\l,left}{\eql@box@ldelim\lbrace}},%
6118 {\r,right}{\eql@box@rdelim\rbrace}},%

```

```
6119    {\eql@decide@true,lr,both}{\eql@box@delim\lbrace\rbrace}}}
```

TODO: describe

```
6120 \eql@define@key{control}{framecell}[\fbox]{%
6121   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
6122 \eql@define@key{control}{frametag}[\fbox]{%
6123   \global\eql@append\eql@tags@container{\def\eql@tags@frame@cmd{#1}}}
```

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

```
6124 \eql@define@key{equations,box}{alt}{}
```

Injections.

```
6125 \eql@define@key{control}{inject}{%
6126   \global\eql@append\eql@interline@container{%
6127     \eql@append\eql@display@injectbefore{#1}}
6128 \eql@define@key{control}{inject*}{%
6129   \global\eql@append\eql@interline@container{%
6130     \eql@append\eql@display@injectafter{#1}}
6131 \eql@define@key{control}{markline}[]{\eql@markline@inject{#1}}
6132 \eql@define@key{control}{markline*}[]{\eql@markline@inject{push,#1}}
6133 \eql@define@key{control}{qed}[]{\eql@markline@inject{qed,#1}}
6134 \eql@define@key{control}{qed*}[]{\eql@markline@inject{qed,push,#1}}}
```

TODO: describe

```
6135 \eql@define@key{markline}{pos}{%
6136   \eql@decide@select{#3}{#2}{#1}{%
6137     {{below,push}{\let\eql@markline@pos\eql@markline@pos@below}},%
6138     {{baseline}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
6139     {{bottom}{\let\eql@markline@pos\eql@markline@pos@bottom}}}}
6140 \eql@define@key{markline}{below,push}[]{%
6141   \let\eql@markline@pos\eql@markline@pos@below}
6142 \eql@define@key{markline}{baseline}[]{%
6143   \let\eql@markline@pos\eql@markline@pos@baseline}
6144 \eql@define@key{markline}{bottom}[]{%
6145   \let\eql@markline@pos\eql@markline@pos@bottom}
6146 \eql@define@key{markline}{shift}{\def\eql@markline@shift{#1}}
6147 \eql@define@key{markline}{symbol}{\def\eql@markline@symbol{#1}}
6148 \eql@define@key{markline}{qed}[]{\let\eql@markline@symbol\eql@markline@qed}
6149 \eql@define@key{setup}{marksymbol}{\def\eql@markline@symbol{#1}}
6150 \eql@define@key{setup}{qedsymbol}{\def\eql@markline@qed{#1}}
6151 \eql@define@key{setup}{markpos}{%
6152   \eql@decide@select{#3}{#2}{#1}{%
6153     {{below}{\let\eql@markline@pos\eql@markline@pos@below}},%
6154     {{baseline}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
6155     {{bottom}{\let\eql@markline@pos\eql@markline@pos@bottom}}}}}
```

Global Switches. Set global switches:

```
6156 \let\eql@multi@linesfallback\eql@false
6157 \let\eql@scan@par\eql@false
6158 \let\eql@single@cr@mode\eql@false
6159 \let\eql@amp@mode\eql@true
6160 \let\eql@ampproof@active\eql@false
```

```

6161 \let\eql@parseopt@warn@env\eql@warn@parseopt
6162 \let\eql@parseopt@warn@cr\@empty

6163 \eql@define@key{equations,setup}{linesfallback}[true]{%
6164   \eql@decide@select{#3}{#2}{#1}{%
6165     {\eql@decide@false{\let\eql@multi@linesfallback\eql@false}},%
6166     {\reuse,lean}{\let\eql@multi@linesfallback\z@}},%
6167     {\measure,full,\eql@decide@true}{\let\eql@multi@linesfallback\eql@true}}}}
6168 \eql@define@key{setup}{ampproof}[true]{%
6169   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
6170 \eql@define@key{equations,setup}{equationcr}[true]{%
6171   \eql@decide@select{#3}{#2}{#1}{%
6172     {\eql@decide@false{\let\eql@single@cr@mode\eql@false}},%
6173     {\eql@decide@true,break}{\let\eql@single@cr@mode\eql@break@cr}},%
6174     {\error,verbose}{\let\eql@single@cr@mode\eql@single@cr@error}}}}
6175 \eql@define@key\eql@keyall{amp}[true]{%
6176   \eql@decide@bool{#3}{#2}{#1}{\eql@amp@mode}}
6177 \eql@define@key\eql@keyall{rescan}[true]{%
6178   \eql@decide@if{#3}{#2}{#1}{%
6179     {\let\eql@scan@body\eql@scan@body@rescan}%
6180     {\let\eql@scan@body\eql@scan@body@dump}}}
6181 \eql@define@key\eql@keyall{scanpar}[true]{%
6182   \eql@decide@bool{#3}{#2}{#1}\eql@scan@par}
6183 \eql@define@key{setup}{defaults}{%
6184   \eql@decide@select{#3}{#2}{#1}{%
6185     {\classic}{\eql@defaults@classic}},%
6186     {\eqnlines}{\eql@defaults@eqnlines}}}}
6187 \eql@define@key\eql@keyall{verbose}[true]{%
6188   \eql@decide@if{#3}{#2}{#1}\eql@verbose@on\eql@verbose@off}

```

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

```

6189 \let\eql@provide@opt@env\tw@
6190 \let\eql@provide@opt@amsmathpatch\eql@false
6191 \let\eql@provide@opt@backup\eql@false
6192 \let\eql@provide@opt@ang\eql@true
6193 \let\eql@provide@opt@eqref\eql@true
6194 \let\eql@provide@opt@matrix\eql@true

6195 \eql@define@key{setup}{amsmathends,amsmathpatch}[true]{%
6196   \eql@error@packageoption{#2}%
6197   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathpatch}
6198 \eql@define@key{setup}{backup}[true]{%
6199   \eql@error@packageoption{#2}%
6200   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
6201 \eql@define@key{setup}{env}[equation]{%
6202   \eql@error@packageoption{#2}%
6203   \eql@decide@select{#3}{#2}{#1}{%
6204     {\none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
6205     {\equation,latex}{\let\eql@provide@opt@env@one}},%
6206     {\amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}}
6207 \eql@define@key{setup}{ang}[true]{%
6208   \eql@error@packageoption{#2}%
6209   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
6210 \eql@define@key{setup}{eqref}[true]{%
6211   \eql@error@packageoption{#2}%
6212   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}
6213 \eql@define@key{setup}{matrix}[true]{%
6214   \eql@error@packageoption{#2}%

```

```
6215 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@matrix}
```

Shortcut Options. **TODO:** describe

```
6216 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
6217 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
6218 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
6219 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
6220 \def\eql@parseopt@eqamp#1{\eqnaddopt{eqamp}\eql@parseopt@peek}
6221 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
6222 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
6223 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
6224 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
6225 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
6226 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
6227 \def\eql@parseopt@punctclear#1{\eql@parseopt@peek~}
6228 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
6229 \def\eql@parseopt@punctnext#1#2{%
6230 \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
6231 \def\eql@parseopt@punctblock#1#2{%
6232 \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
6233 \def\eql@parseopt@punctterm#1{\let\eql@punct@term\eql@true\eql@parseopt@peek}
6234 \def\eql@parseopt@rel symb#1{\eql@parseopt@peek?\eql@class@rel@symb}
6235 \def\eql@parseopt@rel cont#1{\eql@parseopt@peek?{}}
6236 \def\eql@parseopt@rel start#1{\eql@parseopt@peek?relax}
6237 \def\eql@parseopt@rel ord#1{\eql@parseopt@peek?{}}
6238 \def\eql@parseopt@vspace[#1]{%
6239 \advance\eql@vspaceskip@glueexpr#1relax\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 \TeX , \LaTeX and `amsmath` layout closely. These presets mostly use fixed dimensions:

```
6240 \def\eql@defaults@classic{%
6241 \eqnlineset{numberline=all}%
6242 \eqnlineset{condsep={1em}}%
6243 \eqnlineset{matrixsep={2\arraycolsep}}%
6244 \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
6245 \eqnlineset{maxcolsep=off}%
6246 \eqnlineset{spread={\jot}}%
6247 \eqnlineset{tagmargin}%
6248 \eqnlineset{tagmarginratio=1}%
6249 \eqnlineset{tagmarginthreshold=0.5}%
6250 \eqnlineset{leftmargin={\leftmargini}}%
6251 \eqnlineset{padding=max}%
6252 \eqnlineset{evadetag=off}%
6253 \eqnlineset{displayheight=off}%
6254 \eqnlineset{displaydepth=off}%
6255 \eqnlineset{shortmode=belowsingle}%
6256 \eqnlineset{abovecontmode=short}%
6257 \eqnlineset{belowcontmode=short}%
6258 \eqnlineset{aboveparmode=long}%
```

```

6259 \eqnlineset{belowparmode=long}%
6260 \eqnlineset{abovetopmode=long}%
6261 \eqnlineset{belowtopmode=long}%
6262 \eqnlineset{abovelongskip={\abovedisplayskip}}%
6263 \eqnlineset{belowlongskip={\belowdisplayskip}}%
6264 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
6265 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
6266 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
6267 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
6268 \eqnlineset{abovecontskip=0pt}%
6269 \eqnlineset{belowcontskip=0pt}%
6270 \eqnlineset{aboveparskip=0pt}%
6271 \eqnlineset{belowparskip=0pt}%
6272 \eqnlineset{abovetopskip=0pt}%
6273 \eqnlineset{belowtopskip=0pt}%
6274 \eqnlineset{abovetagskip=0pt}%
6275 \eqnlineset{belowtagskip=0pt}%
6276 \eqnlineset{allowbreaks=0}%
6277 \eqnlineset{equationcr=off}%
6278 \eqnlineset{amp=off}%
6279 \eqnlineset{modifier=off}%
6280 \eqnlineset{linesfallback=false}%
6281 \eqnlineset{casesstyle=text}%
6282 \eqnlineset{sqropt=nonumber}%
6283 \eqnlineset{angopt=nonumber}%
6284 }

```

values based on 10pt vs 12pt

`q1@defaults@eqnlines` The (default) preset `eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the `classic` spacing for a 12 pt computer modern font such that 10 pt fonts will lead to slightly reduced spacing. Apart from that, the `eqnlines` setting makes some deliberate layout choices that deviate significantly from `classic` (maximum column separation, no shortening below equations):

```

6285 \def\eq1@defaults@eqnlines{%
6286 \eqnlineset{numberline=all}%
6287 \eqnlineset{condsep=short}%
6288 \eqnlineset{matrixsep=short}%
6289 \eqnlineset{mintagsep=.5em}%
6290 \eqnlineset{maxcolsep=2em}%
6291 \eqnlineset{spread={0.2\normalbaselineskip}}%
6292 \eqnlineset{tagmargin}%
6293 \eqnlineset{tagmarginratio=.334}%
6294 \eqnlineset{tagmarginthreshold=0.5}%
6295 \eqnlineset{leftmargin={\leftmargini}}%
6296 \eqnlineset{padding=0pt}%
6297 \eqnlineset{evadetag}%
6298 \eqnlineset{displayheight=strut}%
6299 \eqnlineset{displaydepth=strut}%
6300 \eqnlineset{shortmode=above}%
6301 \eqnlineset{abovecontmode=noskip}%
6302 \eqnlineset{belowcontmode=long}%
6303 \eqnlineset{aboveparmode=long}%
6304 \eqnlineset{belowparmode=long}%
6305 \eqnlineset{abovetopmode=noskip}%
6306 \eqnlineset{belowtopmode=long}%
6307 \eqnlineset{longskip={0.75\normalbaselineskip}

```

```

6308     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
6309 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
6310     plus 0.25\normalbaselineskip}}%
6311 \eqnlineset{belowshortskip={0.0\normalbaselineskip
6312     plus 0.25\normalbaselineskip}}%
6313 \eqnlineset{medskip={0.4\normalbaselineskip
6314     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
6315 \eqnlineset{abovecontskip=0pt}%
6316 \eqnlineset{belowcontskip=0pt}%
6317 \eqnlineset{aboveparskip=0pt}%
6318 \eqnlineset{belowparskip=0pt}%
6319 \eqnlineset{abovetopskip=0pt}%
6320 \eqnlineset{belowtopskip=0pt}%
6321 \eqnlineset{abovetagskip={0.25\normalbaselineskip
6322     minus 0.25\normalbaselineskip}}%
6323 \eqnlineset{belowtagskip={0.25\normalbaselineskip
6324     minus 0.25\normalbaselineskip}}%
6325 \eqnlineset{allowbreaks=3}%
6326 \eqnlineset{equationcr=break}%
6327 \eqnlineset{amp=on}%
6328 \eqnlineset{modifier=on,modifierenv=off}%
6329 \eqnlineset{linesfallback=true}%
6330 \eqnlineset{casesstyle=false}%
6331 \eqnlineset{sqropt=}
6332 \eqnlineset{angopt=}
6333 }

```

16.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of L^AT_EX 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 6334 \def\eql@provide@movecmd#1#2{%
@provide@undefineenv 6335 \eql@letcs{#1\expandafter}\csname#2\endcsname
6336 }
6337 \def\eql@provide@moveenv#1#2{%
6338 \eql@provide@movecmd{#1}{#2}%
6339 \ifcsname end#2\endcsname
6340 \eql@provide@movecmd{end#1}{end#2}%
6341 \fi
6342 }
6343 \def\eql@provide@undefinecmd#1{%
6344 \eql@letcs{#1}\@undefined
6345 }
6346 \def\eql@provide@undefineenv#1{%
6347 \eql@provide@undefinecmd{#1}%
6348 \eql@provide@undefinecmd{end#1}%
6349 }

```

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:

```
6350 \def\eql@amsmath@endfix#1#2{%
6351   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
6352   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
6353   \eql@provide@movecmd{end#1}{end#2}%
6354   \fi
6355 }
```

`\eql@amsmath@fixmatrix` **TODO:** describe
`\amsmath@fixmatrixend`

```
6356 \def\eql@amsmath@fixmatrix#1{%
6357   \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
6358   \begingroup
6359   \let\matrix@check@gobble
6360   \def\env@matrix{\noexpand\env@matrix}%
6361   \def\env@cases{\noexpand\env@cases}%
6362   \global\edef\eql@tmp{\eql@tmp}%
6363   \endgroup
6364   \eql@letcs{#1}\eql@tmp
6365 }
6366 \def\eql@amsmath@fixmatrixend#1{%
6367   \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
6368   \begingroup
6369   \expandafter\def\expandafter\endmatrix\expandafter{%
6370     \expandafter\unexpanded\expandafter{\endmatrix}}%
6371   \global\long\edef\eql@tmp{\eql@tmp}%
6372   \endgroup
6373   \eql@letcs{end#1}\eql@tmp
6374 }
```

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

```
6375 \def\eql@amsmath@fixends{%
6376   \eql@amsmath@after{%
6377     \eql@amsmath@endfix{flalign}{align}%
6378     \eql@amsmath@endfix{alignat}{align}%
6379     \eql@amsmath@endfix{xalignat}{align}%
6380     \eql@amsmath@endfix{xxalignat}{align}%
6381     \eql@amsmath@endfix{gather*}{gather}%
6382     \eql@amsmath@endfix{multline*}{multline}%
6383     \eql@amsmath@endfix{align*}{align}%
6384     \eql@amsmath@endfix{flalign*}{align}%
6385     \eql@amsmath@endfix{alignat*}{align}%
6386     \eql@amsmath@endfix{xalignat*}{align}%
6387     \eql@amsmath@endfix{gathered}{aligned}%
6388     \eql@amsmath@endfix{alignedat}{aligned}%
6389   }
6390 }
```

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

```

6391 \def\eql@amsmath@fixmatrices{%
6392   \eql@amsmath@after{%
6393     \eql@amsmath@fixmatrix{cases}%
6394     \eql@amsmath@fixmatrix{matrix}%
6395     \eql@amsmath@fixmatrix{pmatrix}%
6396     \eql@amsmath@fixmatrixend{pmatrix}%
6397     \eql@amsmath@fixmatrixend{bmatrix}%
6398     \eql@amsmath@fixmatrixend{Bmatrix}%
6399     \eql@amsmath@fixmatrixend{vmatrix}%
6400     \eql@amsmath@fixmatrixend{Vmatrix}%
6401   }
6402 }

```

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

```

6403 \def\eql@provide@backup@amsenv#1{%
6404   \eql@amsmath@after{%
6405     \eql@provide@moveenv{ams#1}{#1}%
6406     \eql@tagging@register@luamml{ams#1}%
6407     \eql@markline@amsthm@move{ams#1}{#1}%
6408   }%
6409 }

```

`provide@backup@amsbox` **TODO:** describe

```

6410 \def\eql@provide@backup@amsbox#1{%
6411   \eql@amsmath@after{%
6412     \eql@provide@moveenv{ams#1}{#1}%
6413   }%
6414 }

```

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

```

6415 \def\eql@provide@backup@eqref{%
6416   \eql@amsmath@after{%
6417     \eql@provide@movecmd{amseqref}{eqref}%
6418   }%
6419 }

```

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

```

6420 \def\eql@provide@backup@multlined{%
6421   \AddToHook{package/mathtools/after}{%
6422     \eql@provide@moveenv{amsmultlined}{multlined}}%
6423 }

```

`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:

```

6424 \def\eql@provide@backup@equation{%
6425   \eql@amsmath@after{%
6426     \eql@provide@moveenv{amseparation}{equation}%
6427     \eql@provide@moveenv{amseparation*}{equation*}%
6428     \eql@tagging@register@env{amseparation}%
6429     \eql@tagging@register@env{amseparation*}%
6430     \eql@tagging@register@luamml{amseparation}%
6431     \eql@tagging@register@luamml{amseparation*}%
6432     \eql@markline@amsthm@move{amseparation}{equation}%
6433     \eql@markline@amsthm@move{amseparation*}{equation*}%
6434   }%
6435   \AddToHook{package/hyperref/after}{%
6436     \ifpackageloaded{amsmath}{}%
6437     \eql@provide@moveenv{hyperrefequation}{equation}%
6438     \eql@tagging@register@env{hyperrefequation}%
6439     \eql@tagging@register@luamml{hyperrefequation}%
6440     \eql@markline@amsthm@move{hyperequation}{equation}%
6441   }%
6442 }%
6443 \@ifpackageloaded{amsmath}{}%
6444   \@ifpackageloaded{hyperref}{%
6445     \let\latexequation\H@equation
6446     \let\endlatexequation\H@endequation
6447   }\eql@provide@moveenv{latexequation}{equation}}%
6448   \eql@tagging@register@env{latexequation}%
6449   \eql@tagging@register@luamml{latexequation}%
6450   \eql@markline@amsthm@move{latexequation}{equation}%
6451 }%
6452 }

```

`e@backup@displaymath` **TODO:** describe

```

6453 \def\eql@provide@backup@displaymath{%
6454   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
6455   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
6456 }

```

`@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:

```

6457 \def\eql@provide@backup@subequations{%
6458   \eql@amsmath@after{%
6459     \eql@provide@moveenv{amssubequations}{subequations}%
6460   }%
6461   \AddToHook{package/hyperref/after}{%
6462     \AddToHook{cmd/amssubequations/before}{%
6463       {%
6464         \stepcounter{equation}%
6465         \protected@edef\theHparentequation{\theHequation}%
6466         \addtocounter{equation}{-1}%
6467       }%
6468     \AddToHook{cmd/amssubequations/after}{%
6469       {%
6470         \def\theHequation{\theHparentequation\alph{equation}}%
6471         \ignorespaces

```

```

6472   }%
6473 }%
6474 }

```

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

```

6475 \def\eql@provide@backup{%
6476   \eql@provide@backup@eqref
6477   \eql@provide@backup@equation
6478   \eql@provide@backup@displaymath
6479   \eql@provide@backup@amsenv{gather}%
6480   \eql@provide@backup@amsenv{multline}%
6481   \eql@provide@backup@amsenv{align}%
6482   \eql@provide@backup@amsenv{flalign}%
6483   \eql@provide@backup@amsenv{alignat}%
6484   \eql@provide@backup@amsenv{xalignat}%
6485   \eql@provide@backup@amsenv{xxalignat}%
6486   \eql@provide@backup@amsenv{gather*}%
6487   \eql@provide@backup@amsenv{multline*}%
6488   \eql@provide@backup@amsenv{align*}%
6489   \eql@provide@backup@amsenv{flalign*}%
6490   \eql@provide@backup@amsenv{alignat*}%
6491   \eql@provide@backup@amsenv{xalignat*}%
6492   \eql@provide@backup@amsbox{gathered}%
6493   \eql@provide@backup@multlined
6494   \eql@provide@backup@amsbox{aligned}%
6495   \eql@provide@backup@amsbox{alignedat}%
6496   \eql@provide@backup@amsbox{cases}%
6497   \eql@provide@backup@amsbox{matrix}%
6498   \eql@provide@backup@amsbox{pmatrix}%
6499   \eql@provide@backup@amsbox{bmatrix}%
6500   \eql@provide@backup@amsbox{Bmatrix}%
6501   \eql@provide@backup@amsbox{vmatrix}%
6502   \eql@provide@backup@amsbox{Vmatrix}%
6503   \eql@provide@backup@subequations
6504 }

```

Replacement amsmath Environments. **TODO:** describe

```

6505 \def\eql@alignat@gobblecol#1{%
6506   \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.*)

```

6507 \newenvironment{eql@gathered}%
6508   {\eqnadopt{lines}\equationsbox}{\endequationsbox}
6509 \newenvironment{eql@multlined}%
6510   {\eqnadopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
6511 \newenvironment{eql@aligned}%
6512   {\eqnadopt{columns}\equationsbox}{\endequationsbox}
6513 \newenvironment{eql@alignedat}%
6514   {\eqnadopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}%
6515   {\endequationsbox}
6516 \newenvironment{eql@cases}%
6517   {\eqnadopt{cases}\equationsbox}{\endequationsbox}
6518 \newenvironment{eql@matrix}%
6519   {\eqnlineset{modifierboxenv=false}\eqnadopt{matrix=.\}\equationsbox}%

```

```

6520 {\endequationsbox}
6521 \newenvironment{eql@pmatrix}%
6522 {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=r}\equationsbox}%
6523 {\endequationsbox}
6524 \newenvironment{eql@bmatrix}%
6525 {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=s}\equationsbox}%
6526 {\endequationsbox}
6527 \newenvironment{eql@Bmatrix}%
6528 {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=c}\equationsbox}%
6529 {\endequationsbox}
6530 \newenvironment{eql@vmatrix}%
6531 {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=v}\equationsbox}%
6532 {\endequationsbox}
6533 \newenvironment{eql@Vmatrix}%
6534 {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=d}\equationsbox}%
6535 {\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.*)

```

6536 \newenvironment{eql@equation}%
6537 {\eqnaddopt{equation,donumber}\equations}\endequations}
6538 \newenvironment{eql@equation*}%
6539 {\eqnaddopt{equation,nonumber}\equations}\endequations}
6540 \newenvironment{eql@displaymath}%
6541 {\eqnaddopt{equation,nonumber}\equations}\endequations}
6542 \newenvironment{eql@gather}%
6543 {\eqnaddopt{lines,donumber}\equations}\endequations}
6544 \newenvironment{eql@gather*}%
6545 {\eqnaddopt{lines,nonumber}\equations}\endequations}
6546 \newenvironment{eql@multline}%
6547 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,donumber}%
6548 \equations}\endequations}
6549 \newenvironment{eql@multline*}%
6550 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,nonumber}%
6551 \equations}\endequations}
6552 \newenvironment{eql@align}%
6553 {\eqnaddopt{columns,donumber}\equations}\endequations}
6554 \newenvironment{eql@align*}%
6555 {\eqnaddopt{columns,nonumber}\equations}\endequations}
6556 \newenvironment{eql@flalign}%
6557 {\eqnaddopt{columns,fulllength,donumber}\equations}\endequations}
6558 \newenvironment{eql@flalign*}%
6559 {\eqnaddopt{columns,fulllength,nonumber}\equations}\endequations}
6560 \newenvironment{eql@alignat}%
6561 {\eqnaddopt{columns,colsep=off,donumber}%
6562 \eql@alignat@gobblecol\equations}\endequations}
6563 \newenvironment{eql@xalignat}%
6564 {\eqnaddopt{columns,donumber}%
6565 \eql@alignat@gobblecol\equations}\endequations}
6566 \newenvironment{eql@xxalignat}%
6567 {\eqnaddopt{columns,fulllength,donumber}%
6568 \eql@alignat@gobblecol\equations}\endequations}
6569 \newenvironment{eql@alignat*}%
6570 {\eqnaddopt{columns,colsep=off,nonumber}%
6571 \eql@alignat@gobblecol\equations}\endequations}
6572 \newenvironment{eql@xalignat*}%
6573 {\eqnaddopt{columns,nonumber}%

```

```
6574 \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):

```
6575 \def\eql@provide@onlyonce#1#2{%
6576   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
6577   \ifx\eql@tmp\eql@tmpa
6578     \let\eql@tmp\@empty
6579   \fi
6580   \ifx\eql@tmp\@empty
6581     \let\eql@tmp\@undefined
6582     \ifx\@nodocument\relax
6583       \def\eql@tmp{#1}%
6584     \fi
6585     \ifcsname eql@provided@#1\endcsname
6586       \def\eql@tmp{#1}%
6587     \fi
6588     \eql@letcs{eql@provided@#1}\eql@true
6589   \else
6590     \def\eql@tmpa{*}%
6591     \ifx\eql@tmp\eql@tmpa
6592       \def\eql@tmp{#1}%
6593     \fi
6594   \fi
6595 }
```

`\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:

```
6596 \def\eql@provide@eqref#1{%
6597   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
6598   \ifx\eql@tmp\eql@tmpa
6599     \let\eql@tmp\@empty
6600   \fi
6601   \ifx\eql@tmp\@empty
6602     \eql@provide@onlyonce{eqref}{}%
6603   \else
6604     \def\eql@tmpa{*}%
6605     \ifx\eql@tmp\eql@tmpa
6606       \def\eql@tmp{eqref}%
6607     \else
6608       \edef\eql@tmp{\expandafter\@gobble\string#1}%
6609     \fi
6610   \fi
6611   \ifdefined\eql@tmp
6612     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
6613   \else
6614     \eql@amsmath@after{%
```

```

6615     \eql@provide@movecmd{eqref}{eql@eqref}%
6616     }%
6617     \AddToHook{package/mathtools/after}{%
6618         \eql@provide@movecmd{eqref}{eql@eqref}%
6619     }%
6620     \eql@provide@movecmd{eqref}{eql@eqref}%
6621     \eql@amsmath@undefine\eqref
6622     \fi
6623 }

```

`\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:

```

6624 \def\eql@provide@amsenv#1#2{%
6625     \eql@provide@onlyonce{#1}{#2}%
6626     \ifdefined\eql@tmp
6627         \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6628         \eql@tagging@register@luamml{\eql@tmp}%
6629         \eql@markline@amsthm@register{\eql@tmp}%
6630     \else
6631         \eql@amsmath@after{%
6632             \eql@provide@moveenv{#1}{eql@#1}%
6633             \eql@markline@amsthm@register{#1}%
6634         }%
6635         \AddToHook{package/mathtools/after}{%
6636             \eql@provide@moveenv{#1}{eql@#1}%
6637             \eql@markline@amsthm@register{#1}%
6638         }%
6639         \eql@provide@moveenv{#1}{eql@#1}%
6640         \eql@markline@amsthm@register{#1}%
6641         \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6642     \fi
6643 }

```

`\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:

```

6644 \def\eql@provide@amsbox#1#2{%
6645     \eql@provide@onlyonce{#1}{#2}%
6646     \ifdefined\eql@tmp
6647         \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6648     \else
6649         \eql@amsmath@after{%
6650             \eql@provide@moveenv{#1}{eql@#1}}%
6651         \AddToHook{package/mathtools/after}{%
6652             \eql@provide@moveenv{#1}{eql@#1}}%
6653         \eql@provide@moveenv{#1}{eql@#1}%
6654         \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6655     \fi
6656 }

```

`\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:

```

6657 \def\eql@provide@multlined#1{%
6658     \eql@provide@onlyonce{multlined}{#1}%

```

```

6659 \ifdefined\eqL@tmp
6660   \eqL@provide@moveenv{\eqL@tmp}{eqL@multlined}%
6661 \else
6662   \AddToHook{package/mathtools/after}{%
6663     \eqL@provide@moveenv{multlined}{eqL@multlined}}%
6664   \eqL@provide@moveenv{multlined}{eqL@multlined}%
6665   \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
6666     \eqL@provide@undefineenv{multlined}}}%
6667 \fi
6668 }

```

`\eqL@provide@matrix` Provide the cases and matrix environments. Copy into place, and copy again when `amsmath` is loaded:

```

6669 \def\eqL@provide@matrix#1#2#3{%
6670   \eqL@provide@onlyonce{#1}{#3}%
6671   \ifdefined\eqL@tmp
6672     \eqL@provide@moveenv{\eqL@tmp}{eqL@#1}%
6673     \eqL@tagging@register@luamml{\eqL@tmp}%
6674   \else
6675     \eqL@amsmath@after{%
6676       \eqL@provide@moveenv{#1}{eqL@#1}%
6677     }%
6678     \eqL@provide@moveenv{#1}{eqL@#1}%
6679     \ifdefined#2\eqL@amsmath@before{\eqL@provide@undefineenv{#1}}\fi%
6680 \fi
6681 }

```

`\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:

```

6682 \def\eqL@provide@equation#1{%
6683   \eqL@provide@onlyonce{equation}{#1}%
6684   \ifdefined\eqL@tmp
6685     \eqL@provide@moveenv{\eqL@tmp}{eqL@equation}%
6686     \eqL@tagging@register@luamml{\eqL@tmp}%
6687     \eqL@markline@amsthm@register{\eqL@tmp}%
6688   \else
6689     \eqL@amsmath@after{%
6690       \eqL@provide@moveenv{equation}{eqL@equation}%
6691       \eqL@markline@amsthm@register{equation}%
6692     }%
6693     \AddToHook{package/hyperref/after}{%
6694       \ifpackageloaded{amsmath}{%
6695         \eqL@provide@moveenv{equation}{eqL@equation}%
6696         \eqL@markline@amsthm@register{equation}%
6697       }%
6698     }%
6699     \eqL@provide@moveenv{equation}{eqL@equation}%
6700     \eqL@markline@amsthm@register{equation}%
6701     \ifdefined\eqL@tagging@on
6702       \AddToHook{begindocument/end}{%
6703         \eqL@provide@moveenv{equation}{eqL@equation}%
6704         \eqL@markline@amsthm@register{equation}%
6705       }%
6706     \fi
6707 \fi
6708 }

```

`\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:

```

6709 \def\eq@provide@equationstar#1{%
6710   \eq@provide@onlyonce{equation*}{#1}%
6711   \ifdefined\eq@tmp
6712     \eq@provide@moveenv{\eq@tmp}{\eq@equation*}%
6713     \eq@tagging@register@luamml{\eq@tmp}%
6714     \eq@markline@amsthm@register{\eq@tmp}%
6715   \else
6716     \eq@amsmath@after{%
6717       \eq@provide@moveenv{equation*}{\eq@equation*}%
6718       \eq@markline@amsthm@register{equation*}%
6719     }%
6720     \eq@provide@moveenv{equation*}{\eq@equation*}%
6721     \eq@markline@amsthm@register{equation*}%
6722     \eq@amsmath@before{\eq@provide@undefineenv{equation*}}%
6723     \ifdefined\eq@tagging@on
6724       \AddToHook{begindocument/end}{%
6725         \eq@provide@moveenv{equation*}{\eq@equation*}%
6726         \eq@markline@amsthm@register{equation*}%
6727       }%
6728     \fi
6729   \fi
6730 }

```

`\provide@displaymath` **TODO:** describe

```

6731 \def\eq@provide@displaymath#1{%
6732   \eq@provide@onlyonce{displaymath}{#1}%
6733   \ifdefined\eq@tmp
6734     \eq@provide@moveenv{\eq@tmp}{\eq@displaymath}%
6735     \eq@markline@amsthm@register{\eq@tmp}%
6736     \eq@tagging@register@luamml{\eq@tmp}%
6737   \else
6738     \eq@provide@moveenv{displaymath}{\eq@displaymath}%
6739     \eq@markline@amsthm@register{displaymath}%
6740     \ifdefined\eq@tagging@on
6741       \AddToHook{begindocument/end}{%
6742         \eq@provide@moveenv{displaymath}{\eq@displaymath}}%
6743     \fi
6744   \fi
6745 }

```

`\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:

```

6746 \def\eq@provide@subequations#1{%
6747   \eq@provide@onlyonce{subequations}{#1}%
6748   \ifdefined\eq@tmp
6749     \eq@provide@moveenv{\eq@tmp}{\eq@subequations}%
6750   \else

```

```

6751 \eql@amsmath@after{%
6752   \eql@provide@moveenv{subequations}{eql@subequations}%
6753 }%
6754 \AddToHook{package/hyperref/after}{%
6755   \AddToHook{cmd/subequations/before}[hyperref]{}%
6756   \AddToHook{cmd/subequations/after}[hyperref]{}%
6757   \RemoveFromHook{cmd/subequations/before}[hyperref]%
6758   \RemoveFromHook{cmd/subequations/after}[hyperref]%
6759   \AddToHook{begindocument/end}{%
6760     \eql@provide@moveenv{subequations}{eql@subequations}}%
6761 }%
6762 \eql@provide@moveenv{subequations}{eql@subequations}%
6763 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6764 \fi
6765 }

```

`\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:

```

6766 \def\eql@provide@sqr{%
6767   \let\[ \eql@sqr@open
6768   \let\] \eql@sqr@close
6769   \eql@amsmath@after{%
6770     \let\[ \eql@sqr@open
6771     \let\] \eql@sqr@close
6772   }%
6773   \ifdefined\eql@tagging@on
6774     \AddToHook{begindocument/end}{%
6775       \let\[ \eql@sqr@open
6776       \let\] \eql@sqr@close
6777     }%
6778   \fi
6779 }

```

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

```

6780 \def\eql@provide@ang{%
6781   \let\< \eql@ang@open
6782   \let\> \eql@ang@close
6783 }

```

TODO: describe

```

6784 \def\eql@provide@tagform{%
6785   \def\tagform##1{\maketag@@{\eql@tags@tagform{##1}}}
6786 \def\eql@provide@maketag{%
6787   \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:

```

6788 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
6789 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
6790 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
6791 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}

```

```

6792 \eql@define@key{provide}{multiline} [] {\eql@provide@amsenv{multiline}{#1}}
6793 \eql@define@key{provide}{align} [] {\eql@provide@amsenv{align}{#1}}
6794 \eql@define@key{provide}{flalign} [] {\eql@provide@amsenv{flalign}{#1}}
6795 \eql@define@key{provide}{alignat} [] {\eql@provide@amsenv{alignat}{#1}}
6796 \eql@define@key{provide}{xalignat} [] {\eql@provide@amsenv{xalignat}{#1}}
6797 \eql@define@key{provide}{xxalignat} [] {\eql@provide@amsenv{xxalignat}{#1}}
6798 \eql@define@key{provide}{gather*} [] {\eql@provide@amsenv{gather*}{#1}}
6799 \eql@define@key{provide}{multiline*} [] {\eql@provide@amsenv{multiline*}{#1}}
6800 \eql@define@key{provide}{align*} [] {\eql@provide@amsenv{align*}{#1}}
6801 \eql@define@key{provide}{flalign*} [] {\eql@provide@amsenv{flalign*}{#1}}
6802 \eql@define@key{provide}{alignat*} [] {\eql@provide@amsenv{alignat*}{#1}}
6803 \eql@define@key{provide}{xalignat*} [] {\eql@provide@amsenv{xalignat*}{#1}}
6804 \eql@define@key{provide}{gathered} [] {\eql@provide@amsbox{gathered}{#1}}
6805 \eql@define@key{provide}{multlined} [] {\eql@provide@multlined{#1}}
6806 \eql@define@key{provide}{aligned} [] {\eql@provide@amsbox{aligned}{#1}}
6807 \eql@define@key{provide}{alignedat} [] {\eql@provide@amsbox{alignedat}{#1}}
6808 \eql@define@key{provide}{cases} [] {\eql@provide@matrix{cases}\eql@false{#1}}
6809 \eql@define@key{provide}{matrix} [] {\eql@provide@matrix{matrix}\eql@false{#1}}
6810 \eql@define@key{provide}{pmatrix} [] {\eql@provide@matrix{pmatrix}\eql@false{#1}}
6811 \eql@define@key{provide}{bmatrix} [] {\eql@provide@matrix{bmatrix}\eql@true{#1}}
6812 \eql@define@key{provide}{Bmatrix} [] {\eql@provide@matrix{Bmatrix}\eql@true{#1}}
6813 \eql@define@key{provide}{vmatrix} [] {\eql@provide@matrix{vmatrix}\eql@true{#1}}
6814 \eql@define@key{provide}{Vmatrix} [] {\eql@provide@matrix{Vmatrix}\eql@true{#1}}
6815 \eql@define@key{provide}{subequations} [] {\eql@provide@subequations{#1}}
6816 \eql@define@key{provide}{sqr} [] {\eql@provide@sqr}
6817 \eql@define@key{provide}{ang} [] {\eql@provide@ang}
6818 \eql@define@key{provide}{eqref} [] {\eql@provide@eqref{#1}}
6819 \eql@define@key{provide}{tagform} [] {\eql@provide@tagform}
6820 \eql@define@key{provide}{maketag} [] {\eql@provide@maketag}

```

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

```

6821 \newcommand{\eqnlinesprovide}[1]{%
6822   \eql@setkeys{provide}{#1}%
6823   \ignorespaces
6824 }

```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6825 \let\eql@error@packageoption\@gobble
```

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

```

6826 \DeclareOption{leqno}{\eqnlinesset{tagsleft}}
6827 \DeclareOption{fleqn}{\eqnlinesset{left}}

```

Pass undeclared options on to keyval processing:

```
6828 \DeclareOption*{\expandafter\eqnlinesset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

6829 \eql@defaults@eqnlines
6830 \eql@equations@columns@set
6831 \eql@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:

```
6832 \ifdefined\tagsleft@true\else
6833   \expandafter\newif\csname iftagsleft@\endcsname
6834 \fi
6835 \ifdefined@\fleqntrue\else
6836   \expandafter\newif\csname if@fleqn\endcsname
6837 \fi
```

Import amsmath switches `leqno` as `tagsleft` and `fleqn` as `left`:

```
6838 \eql@amsmath@after{%
6839   \ifnum\eql@provide@opt@env=\tw@
6840     \iftagsleft@
6841       \eqnlineset{tags=left}%
6842     \else
6843       \eqnlineset{tags=right}%
6844     \fi
6845   \if@fleqn
6846     \eqnlineset{layout=left}%
6847   \else
6848     \eqnlineset{layout=center}%
6849   \fi
6850 \fi
6851 }
```

Process package options:

```
6852 \ProcessOptions
```

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

```
6853 \def\eql@error@packageoption#1{%
6854   \eq@error{may only use '#1' as a package option}%
6855 }
```

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

```
6856 \ifnum\eql@provide@opt@env=\tw@
6857 \ifdefined\eql@provide@opt@matrix
6858   \let\eql@provide@opt@amsmathpatch\eql@false
6859 \fi\fi
6860 \ifdefined\eql@provide@opt@backup
6861   \let\eql@provide@opt@amsmathpatch\eql@true
6862 \fi
6863 \ifdefined\eql@provide@opt@amsmathpatch
6864   \eq@amsmath@fixends
6865   \eq@amsmath@fixmatrices
6866 \fi
```

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

```
6867 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

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

```
6868 \ifnum\eql@provide@opt@env>\z@
6869   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6870 \fi
```

Provide amsmath equation environments if desired:

```
6871 \ifnum\eq@provide@opt@env=\tw@
6872 \eqnlinesprovide{%
6873     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6874     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6875     multlined,gathered,aligned,alignedat,%
6876     subequations}
6877 \fi
```

Provide symbolic shortcut $\langle \dots \rangle$ if desired:

```
6878 \ifdefined\eq@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference \eqref if desired:

```
6879 \ifdefined\eq@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```

Provide cases and matrix environments if desired:

```
6880 \ifdefined\eq@provide@opt@matrix
6881 \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
6882 \fi
```