

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`

2026/02/07, v0.15

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

Abstract

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

Contents

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

4.2	Math Classes at Alignment	25
4.3	Framed Cells	26
4.4	Single-Line Composition	27
4.5	Alternative Content Description	29
5	Equation Numbering	29
5.1	Supporting Definitions	29
5.2	Schemes	30
5.3	Interface	34
5.4	Integration	39
5.5	Positioning	42
5.6	Component Display	45
5.7	Tag Composition	48
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	54
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	72
9.1	Supporting Definitions	72
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	81
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	100
11.6	Main	102

11.7 Options Processing	104
11.8 Environment	105
12 Single-Line Equation	107
12.1 Native Mode	107
12.2 Print	107
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	112
14.1 Support	113
14.2 Column Breaks	113
14.3 Transpose	114
14.4 Measure	116
14.5 Columns Placement	119
14.6 Print	119
15 Interface	123
15.1 Options Processing	123
15.2 Single-Line Main	125
15.3 Multi-Line Main	126
15.4 Equations Environment	128
16 Options	131
16.1 Selection Tools	131
16.2 Options Declarations	132
16.3 Parameter Presets	147
16.4 Component Selection	149
16.5 Global and Package Options	160

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@sregroup`:

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

2.5 Error Messages

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

```

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

```

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

```

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

```

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

```

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

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

```

2.6 amsmath Integration

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

```

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

```

TODO: temporary fix for development stages

```

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

```

2.7 PDF Tagging Support

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

```

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

```

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


```

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

```

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

```

196   \let\eql@single@doscan\eql@true
197   \let\eql@scan@body\eql@scan@body@rescan

```

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

```

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

```

When tagging is deactivated, provide empty definitions:

```

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

```

2.8 Key-Value Processing

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

```

233 \RequirePackage{keyval}

```

Value Selection.

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

$$\{ \{ \{ val1a, val1b, \dots \} \{ act1 \} \}, \{ \{ val2a, val2b, \dots \} \{ act2 \} \}, \dots \}. \quad (1)$$

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

```

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

```

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

```

Options and Control Interface.

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

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

`\eqnaddopt`

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

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

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

`\eql@control@default`

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

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

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

2.9 Scanning the Equation Body

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

Token Register.

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

```
316 \newtoks\eql@scan@reg@
```

`\eql@scan@body@dump` The macro `\eql@scan@body@dump` dumps the equation body from the register so that we do not have to pass it around in arguments. The macro `\eql@scan@body@rescan` rescans the tokens so that special commands such as `\verb` can be processed properly. The register `\eql@scan@body` holds the currently selected mode of operation: **TODO**: may skip `\expandafter` before `\scantokens`?

```

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

```

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

```

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

```

Environment Body. The following mechanism scans the contents of an environment taking into account nested environments that may be contained in the body.

`\eql@scan@env` The macro `\eql@scan@env` starts the scan for the `\end{...}` command of the current environment. The argument is a call-back macro to process the body in `\eql@scan@reg@`:

```

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

```

We call `\eql@scan@env@iterate` which will scan until the next occurrence of `\end` and then count the number of occurrences of `\begin` before `\end` in `\eql@scan@stack`. If we simply called `\eql@scan@env@iterate` directly, the error message for an unwanted `\par` token (usually from a blank line) would refer to `\eql@scan@env@iterate` which would not be illuminating. We use a little finesse to get a more intelligible error message: We use the actual environment name as the name of the temporary function that is `\let` to `\eql@scan@env@iterate`:

```

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

```

`\eql@scan@env@iterate` `\eql@scan@env@iterate` takes two arguments: the first will consist of all text up to the next `\end` command, the second will be the `\end` command's argument. If there are any extra `\begin` commands in the body text, a marker is pushed onto a stack via `\eql@scan@env@count`. An empty state for this stack means that we have reached the `\end` that matches our original `\begin`. Otherwise we need to include the `\end` and its argument in the material that we are adding to our environment body accumulator:

```

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

```

```

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

```

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

```

349 \def\eql@scan@env@iterate@nopar#1\end#2{\eql@scan@env@iterate#1\end{#2}}

```

`\eql@scan@env@count` When adding a piece of the current environment's contents to `\eql@scan@reg@`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

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

```

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

```

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

```

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

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Interface. TODO: describe

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

```

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

```

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

```

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

```

`\eql@sqr@cancel`

```

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

```

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

```

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

```

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

```

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

```

3 Parameters and Registers

In the following, we collect parameter and register definitions.

3.1 Parameters

TODO: describe

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

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

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

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

```
442 \let\eql@layoutleft\eql@false
```

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

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

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

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

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

```
447 \newdimen\eql@tagmargin@
448 \let\eql@tagmargin@val\@undefined
449 \newdimen\eql@tagmargin@ratio@
450 \eql@tagmargin@ratio@\p@
451 \def\eql@tagmargin@threshold{0.5}
```

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

```
452 \newdimen\eql@indent@
453 \def\eql@indent@val{2em}
```

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

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

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

`\eqldisplay@marginleft`

`\eqldisplay@marginright`

```

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

```

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

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

`\eql@box@condsep`

```

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

```

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

`\eql@break@line@shortsep`

`\eql@break@col@sep`

`\eql@break@col@shortsep`

```

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

```

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

```

468 \let\eql@spread@reset\eql@false
469 \def\eql@spread@val{\jot}
470 \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 \TeX , nevertheless: **TODO:** probably do not need this due to fixed point arithmetic.

```

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

```

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

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

```

473 \def\eqldisplay@height\@undefined
474 \def\eqldisplay@depth\@undefined

```

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

```

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

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

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

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

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

```

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

```

\eq@skip@force@below
\mode@above@ (counter) 484 \newcount\eq@skip@mode@above@
\mode@below@ (counter) 485 \newcount\eq@skip@mode@below@
486 \let\eq@skip@force@above@\undefined
487 \let\eq@skip@force@below@\undefined
488 \let\eq@skip@custom@above@\undefined
489 \let\eq@skip@custom@below@\undefined

```

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

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

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

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

```

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

```

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

```

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

```

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

```

507 \newdimen\eq@tagwidthmin@
508 \eq@tagwidthmin@\z@

```

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

```

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

```

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

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

```
\eql@equations@ang@opt
\eq1@box@ang@opt
511 \def\eq1@equations@sqr@opt{equation}
512 \def\eq1@equations@ang@opt{align}
513 \def\eq1@box@ang@opt{align}
```

Multi-Line Align Mode.

```
514 \let\eq1@columns@fulllength\eq1@false
```

3.2 Registers

TODO: describe

General. **TODO:** describe

```
515 \newcount\eq1@count@
516 \newdimen\eq1@dimen@
517 \newskip\eq1@skip@
```

TODO: describe

```
518 \let\eq1@display@container\@empty
```

`\eq1@cellbox@` (*box*) The box `\eq1@cellbox@` holds the present alignment component and `\eq1@tagbox@` the
`\eq1@tagbox@` (*box*) tag for the present line. The corresponding dimensions `\eq1@cellwidth@` and
`\eq1@cellwidth@` (*dimen*) `\eq1@tagwidth@` hold their widths. `\eq1@prevwidth@` holds the width of the previous
`\eq1@prevwidth@` (*dimen*) alignment component: **TODO:** adjust

```
\eq1@tagwidth@ (dimen)
\eq1@prevdepth@ (dimen)
\eq1@prevgraf@ (counter)
519 \newbox\eq1@cellbox@
520 \newbox\eq1@tagbox@
521 \newdimen\eq1@cellwidth@
522 \newdimen\eq1@prevwidth@
523 \newdimen\eq1@tagwidth@
524 \newdimen\eq1@prevdepth@
525 \newcount\eq1@prevgraf@
```

```
\eq1@totalwidth@ (dimen)
\eq1@tagwidth@max@ (dimen)
\eq1@totalheight@ (dimen)
526 \newdimen\eq1@totalwidth@
527 \newdimen\eq1@tagwidth@max@
528 \newdimen\eq1@totalheight@
529 \newdimen\eq1@topheight@
530 \newdimen\eq1@bottomdepth@
```

`\eq1@line@height@` (*dimen*) The dimension registers `\eq1@line@height@` and `\eq1@line@depth@` keep track of the
`\eq1@line@depth@` (*dimen*) height and depth of the present line in an alignment:

```
531 \newdimen\eq1@line@height@
532 \newdimen\eq1@line@depth@
```

```
\eq1@line@width@ (dimen)
\eq1@line@avail@ (dimen)
\eq1@line@pos@ (dimen)
\eq1@widthsep@ (counter)
\eq1@availsep@ (counter)
\eq1@line@possep@ (counter)
\eq1@line@offset@ (dimen)
\eq1@prevdepth@ (dimen)
\eq1@interline@ (dimen)
533 \newdimen\eq1@line@width@
534 \newdimen\eq1@line@avail@
535 \newdimen\eq1@line@pos@
```

```

536 \newcount\eql@line@availsep@
537 \newcount\eql@line@widthsep@
538 \newcount\eql@line@possep@
539 \newdimen\eql@line@offset@
540 \newdimen\eql@line@prevdepth@
541 \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*)

```

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

```

`\eql@column@`

`\eql@totalcolumns@`

```

545 \newcount\eql@column@
546 \newcount\eql@totalcolumns@

```

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

```

547 \newdimen\eql@colsep@

```

`\intercolumns@` (*counter*)

```

548 \newcount\eql@intercolumns@

```

Vertical Spacing Adjustments.

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

```

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

```

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

```

556 \newdimen\eql@display@aboveextend@
557 \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:

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

`\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:

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

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

4 Features

4.1 Punctuation

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

`\eql@punct@col` These macros store the punctuation character for columns, lines and blocks. An undefined value indicates that the punctuation should be handed down to the next lower level:
`\eql@punct@line`
`\eql@punct@block` **TODO:** update
`\eql@punct@next`
`\eql@punct@top` 583 \let\eql@punct@col\@empty

```

584 \let\eqlopunct@line\@undefined
585 \let\eqlopunct@block\@undefined
586 \let\eqlopunct@next\@undefined
587 \let\eqlopunct@top\@undefined

```

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

```

588 \let\eqlopunct@class\@empty
589 \let\eqlopunct@sep\@empty

```

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

```

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

```

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

```

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

```

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

TODO: describe

```

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

```

TODO: describe

```

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

```

`\eqnpunct` **TODO:** describe

623 \let\eqnpunct\eql@punct@adopt

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

```

624 \def\eql@punct@fill@next#1{%
625   \ifdefined\eql@punct@next
626     \ifx\eql@punct@next\relax
627       \let\eql@punct@next\@undefined
628     \fi
629   \else
630     \ifx\eql@punct@top\relax\else
631       \let\eql@punct@next#1%
632     \fi
633   \fi
634 }
```

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

```

635 \def\eql@punct@output@next{%
636   \ifx\eql@punct@next\@empty\else
637     \ifmmode\eql@punct@class\fi
638     \eql@punct@sep
639     \eql@punct@next
640   \fi
641   \let\eql@punct@next\@undefined
642 }
```

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

```

643 \def\eql@punct@print@next{%
644   \ifdefined\eql@punct@next
645     \eql@punct@output@next
646   \fi
647 }
```

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

```

648 \def\eql@punct@apply@next{%
649   \ifdefined\eql@punct@next
650     \eql@punct@output@next
651     \eql@punct@top@stop
652   \fi
653 }
```

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

```

654 \def\eql@punct@print@col{%
655   \eql@punct@fill@next\eql@punct@col
656   \eql@punct@print@next
657 }
```

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

```

658 \def\eql@punct@apply@col{%
659   \eql@punct@fill@next\eql@punct@col
660   \eql@punct@apply@next
661 }
```

Output the punctuation for the present line unless disabled:

`\eql@punct@print@line`

```
662 \def\eql@punct@print@line{%
663   \eql@punct@fill@next\eql@punct@line
664   \eql@punct@print@next
665 }
```

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

`\eql@punct@apply@line`

```
666 \def\eql@punct@apply@line{%
667   \eql@punct@fill@next\eql@punct@line
668   \eql@punct@apply@next
669 }
```

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

```
670 \def\eql@punct@apply@block{%
671   \eql@punct@fill@next\eql@punct@block
672   \eql@punct@apply@next
673 }
```

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

```
674 \def\eql@punct@apply@top{%
675   \eql@punct@fill@next\eql@punct@top
676   \eql@punct@print@next
677   \global\eql@punct@top@stop
678 }
```

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

```
679 \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.

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

```
680 \def\eql@class@innerright@sel@{%
681   \ifdim\eql@prevwidth@=\z@
682     \eql@class@innerlead
683   \else
684     \eql@class@innerright
685   \fi
686 }
```

`\@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:

`\@class@innerright@set`

`\@class@innerlead@set`

```
687 \def\eql@class@innerleft@set#1{%
688   \def\eql@class@innerleft{#1}%
689 }
```

```

690 \def\eq@class@innerright@set#1{%
691   \def\eq@class@innerright{#1}%
692   \let\eq@class@innerright@sel\eq@class@innerright
693 }
694 \def\eq@class@innerlead@set#1{%
695   \def\eq@class@innerlead{#1}%
696   \let\eq@class@innerright@sel\eq@class@innerright@sel@
697 }

698 \def\eq@class@rel@symb{=}
699 \def\eq@class@rel@cont#1{\eq@class@rel@start{#1}\mathclose{}}
700 \def\eq@class@rel@start#1{\mathrel{\phantom{#1}}}
701 \def\eq@class@rel@amp{&}
702 \def\eq@class@rel@amprelax{&\relax}
703 \def\eq@class@rel@relaxamp{\relax&}
704 \def\eq@class@rel@make#1{%
705   \ifdefined\eq@class@rel@after
706     \def\eq@tmp{&#1}%
707     \ifx\eq@tmp\eq@class@rel@amp
708       \def\eq@tmp{&\eq@class@rel@cont\eq@class@rel@symb}%
709     \else\ifx\eq@tmp\eq@class@rel@amprelax
710       \def\eq@tmp{&\eq@class@rel@start\eq@class@rel@symb}%
711     \fi\fi
712   \else
713     \def\eq@tmp{#1&%}
714     \ifx\eq@tmp\eq@class@rel@relaxamp
715       \def\eq@tmp{\hskip1sp&%}
716     \fi
717   \fi
718   \eq@tmp
719 }

```

`\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`):

```

720 \def\eq@class@ampeq{%
721   \eq@class@innerleft@set{}%
722   \eq@class@innerright@set{{}}%
723   \let\eq@class@rel@after\eq@true
724 }
725 \def\eq@class@eqamp{%
726   \eq@class@innerleft@set{\mathrel{}}%
727   \eq@class@innerright@set{\mathrel{}}%
728   \eq@class@innerlead@set{{}}%
729   \let\eq@class@rel@after\eq@false
730 }
731 \eq@class@ampeq

```

4.3 Framed Cells

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

```

732 \let\eq@frame@cmd\@undefined
733 \newdimen\eq@frame@margin@
734 \def\eq@frame@set[#1]{%
735   \global\eq@append\eq@cell@container{\def\eq@frame@cmd{#1}}
736   \protected\def\framecell{\eq@testopt@tight@ampsafe\eq@frame@set\fbx}
737 \def\eq@frame@measure{%

```

```

738 \setbox\z@\hbox{\eql@frame@cmd{}}}%
739 \eql@frame@margin@.5\wd\z@
740 }
741 \def\eql@frame@print{%
742 \setbox\eql@cellbox@\hbox{%
743 \eql@frame@cmd{\unhbox\eql@cellbox@}%
744 }%
745 }
746 \def\eql@frame@adjust{%
747 \setbox\eql@cellbox@\hbox{%
748 \eql@frame@cmd{%
749 \unhbox\eql@cellbox@
750 \unkern
751 \unskip
752 }%
753 \hfil
754 \kern\z@
755 }%
756 }

```

4.4 Single-Line Composition

TODO: describe

```

\eql@break@line
\eql@break@col
757 \def\eql@break@line{%
758 \let\eql@break@sep\eql@break@line@sep
759 \let\eql@break@shortsep\eql@break@line@shortsep
760 \let\eql@break@print\eql@punct@print@line
761 \let\eql@punct@term\eql@false
762 \let\eql@class@rel@composed\@undefined
763 \eql@ampprotect\eql@break@test\eql@break@process}
764 \def\eql@break@col{%
765 \let\eql@break@sep\eql@break@col@sep
766 \let\eql@break@shortsep\eql@break@col@shortsep
767 \let\eql@break@print\eql@punct@print@col
768 \let\eql@punct@term\eql@false
769 \let\eql@class@rel@composed\@undefined
770 \eql@ampprotect\eql@break@test\eql@break@process}
771 \def\eql@break@cr{%
772 \let\eql@break@sep\eql@break@line@sep
773 \let\eql@break@shortsep\eql@break@line@shortsep
774 \let\eql@break@print\eql@punct@print@line
775 \let\eql@punct@term\eql@false
776 \let\eql@class@rel@composed\@undefined
777 \eql@ampprotect\eql@break@cr@test\eql@break@process}
778 \def\eql@break@amp{%
779 \eql@ampprotecttwo\eql@break@amp@testescape
780 \eql@amp@org\eql@break@process}
781 \def\eql@break@amp@testescape#1#2{%
782 \eql@ifnextgobble@tight/{#1}{\eql@break@amp@testnoescape{#2}}}
783 \def\eql@break@amp@testnoescape#1{%
784 \relax
785 \let\eql@break@sep\eql@break@col@sep
786 \let\eql@break@shortsep\eql@break@col@shortsep
787 \let\eql@break@print\eql@punct@print@col
788 \let\eql@punct@term\eql@false

```

```

789 \let\eql@class@rel@composed\undefined
790 \eql@break@amp@test{#1}}

```

TODO: describe

```

791 \def\eql@break@test@setopt{\let\eql@break@test\eql@break@testopt}
792 \def\eql@break@test@setall{\let\eql@break@test\eql@break@testall}
793 \def\eql@break@cr@test@setopt{\let\eql@break@cr@test\eql@break@testopt}
794 \def\eql@break@cr@test@setall{\let\eql@break@cr@test\eql@break@testall}
795 \def\eql@break@amp@test@setopt{\let\eql@break@amp@test\eql@break@testopt}
796 \def\eql@break@amp@test@setall{\let\eql@break@amp@test\eql@break@testall}

```

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

```

797 \def\eql@break@testopt#1{\eql@ifstar@tight
798   {\let\eql@break@sep\eql@break@shortsep#1}{\eql@break@testopt@arg{#1}}}
799 \def\eql@break@testopt@arg#1{\eql@ifnextchar@tight [%
800   {\eql@break@testopt@set{#1}}{#1}}
801 \def\eql@break@testopt@set#1[#2]{\def\eql@break@sep{#2}#1}

```

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

`@break@testall@parse`

```

802 \def\eql@break@testall{\eql@parseopt@cr\eql@break@testall@parse}
803 \def\eql@break@testall@parse{%
804   \ifx\eql@parseopt@token[%
805     \let\eql@parseopt@next\eql@break@parse@val
806   \fi
807   \ifx\eql@parseopt@token*%
808     \let\eql@parseopt@next\eql@break@parse@star
809   \fi
810   \ifx\eql@parseopt@token.%
811     \let\eql@parseopt@next\eql@parseopt@punctpass
812   \fi
813   \ifx\eql@parseopt@token,%
814     \let\eql@parseopt@next\eql@parseopt@punctpass
815   \fi
816   \ifx\eql@parseopt@token~%
817     \let\eql@parseopt@next\eql@parseopt@punctpass
818   \fi
819   \ifx\eql@parseopt@token'%
820     \let\eql@parseopt@next\eql@parseopt@punctnext
821   \fi
822   \ifx\eql@parseopt@token!%
823     \let\eql@parseopt@next\eql@parseopt@punctterm
824   \fi
825   \ifx\eql@parseopt@token/%
826     \let\eql@parseopt@next\eql@parseopt@punctclear
827   \fi
828   \ifx\eql@parseopt@token=%
829     \let\eql@parseopt@next\eql@parseopt@relsymp
830   \fi
831   \ifx\eql@parseopt@token;%
832     \let\eql@parseopt@next\eql@parseopt@relcont
833   \fi
834   \ifx\eql@parseopt@token:%
835     \let\eql@parseopt@next\eql@parseopt@relstart
836   \fi
837   \ifx\eql@parseopt@token|%
838     \let\eql@parseopt@next\eql@break@parse@rel

```

```

839 \fi
840 }
841 \def\eql@break@parse@val[#1]{%
842 \def\eql@break@sep{#1}\eql@parseopt@peek}
843 \def\eql@break@parse@star#1{%
844 \let\eql@break@sep\eql@break@shortsep\eql@parseopt@peek}
845 \def\eql@break@parse@rel#1#2{%
846 \def\eql@class@rel@composed{#2}\eql@parseopt@end}

```

`\eql@break@process`

```

847 \def\eql@break@process{%
848 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
849 \ifdefined\eql@class@rel@composed
850 \eql@class@rel@composed
851 \else
852 \eql@break@print
853 \hspace{\glueexpr\eql@break@sep\relax}%
854 \fi
855 }

```

`\eql@break@join`

```

856 \def\eql@break@join{\eql@srbgroup
857 \eql@ifstar@tight
858 {\eql@break@join@opt[\eql@break@col@shortsep]}%
859 {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
860 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
861 \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

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

```

862 \newcommand{\eqnsep}{\eql@break@col}
863 \newcommand{\eqnbreak}{\eql@break@line}
864 \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

```

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

```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```

866 \let\eql@tags@autolabel\eql@false
867 \let\eql@tags@autotag\eql@true
868 \let\eql@tags@warn\eql@true

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

```

```

870 \let\eql@tagpos@doconvert\eql@false
871 \def\eql@tagpos@snap{4pt}

```

Registers.

```

872 \let\eql@numbering@mode\@undefined

873 \let\eql@numbering@active\eql@true
874 \let\eql@numbering@multi\eql@true

875 \let\eql@tags@container\@undefined
876 \def\eql@tags@container@clear{%
877   \let\eql@tags@label\@undefined
878   \let\eql@tags@name\@undefined
879   \let\eql@tags@tag\@undefined
880   \let\eql@tags@ref\@undefined
881   \let\eql@tags@anchor\@empty
882   \eql@tagpos@shift@z@
883   \eql@tagpos@smashup@z@
884   \eql@tagpos@smashdown@z@
885   \let\eql@tagpos@reserve\eql@true
886 }

887 \let\eql@tags@label\@undefined
888 \let\eql@tags@name\@undefined
889 \let\eql@tags@tag\@undefined
890 \let\eql@tags@ref\@undefined
891 \let\eql@tags@frame@cmd\@firstofone

```

`\eql@tags@glabel@` (*counter*)

```

892 \newcount\eql@tags@glabel@
893 \eql@tags@glabel@z@
894 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
895 \def\eql@tags@glabel@step{\global\advance\eql@tags@glabel@\@ne}

896 \let\eql@tagpos@continuous\eql@false

897 \newcount\eql@tagpos@row@
898 \newcount\eql@tagpos@prevrow@
899 \newdimen\eql@tagpos@shift@
900 \newdimen\eql@tagpos@smashup@
901 \newdimen\eql@tagpos@smashdown@
902 \newdimen\eql@tagpos@current@
903 \newdimen\eql@tagpos@plain@
904 \newdimen\eql@tagpos@raised@
905 \newdimen\eql@tagpos@target@
906 \newdimen\eql@tagpos@headroom@
907 \newdimen\eql@tagpos@footroom@

```

5.2 Schemes

TODO: describe

Table.

```

908 \def\eql@numbering@tab@sub{sub}
909 \def\eql@numbering@tab@all{all}
910 \def\eql@numbering@tab@first{first}
911 \def\eql@numbering@tab@last{last}
912 \def\eql@numbering@tab@in{in}
913 \def\eql@numbering@tab@out{out}
914 \def\eql@numbering@tab@middle{middle}
915 \def\eql@numbering@tab@best{best}
916 \def\eql@numbering@tab@here{here}
917 \def\eql@numbering@tab@top{top}
918 \def\eql@numbering@tab@bottom{bottom}
919 \def\eql@numbering@tab@center{center}
920 \def\eql@numbering@tab@centerone{centerone}
921 \def\eql@numbering@tab@median{median}
922 \def\eql@numbering@tab@baseline{baseline}

923 \let\eql@numbering@mode\eql@numbering@tab@all
924 \let\eql@numbering@mode@multi\eql@numbering@tab@all
925 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```

926 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
927 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
928 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
929 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
930 \let\eql@numbering@tab@outside\eql@numbering@tab@out
931 \let\eql@numbering@tab@inside\eql@numbering@tab@in
932 \let\eql@numbering@tab@within\eql@numbering@tab@in
933 \let\eql@numbering@tab@opt\eql@numbering@tab@best
934 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
935 \let\eql@numbering@tab@pick\eql@numbering@tab@here
936 \let\eql@numbering@tab@med\eql@numbering@tab@median
937 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
938 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone

```

TODO: describe

```

939 \let\eql@numbering@tab@a\eql@numbering@tab@all
940 \let\eql@numbering@tab@s\eql@numbering@tab@sub
941 \let\eql@numbering@tab@f\eql@numbering@tab@first
942 \let\eql@numbering@tab@l\eql@numbering@tab@last
943 \let\eql@numbering@tab@o\eql@numbering@tab@out
944 \let\eql@numbering@tab@i\eql@numbering@tab@in
945 \let\eql@numbering@tab@h\eql@numbering@tab@here
946 \let\eql@numbering@tab@t\eql@numbering@tab@top
947 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
948 \let\eql@numbering@tab@c\eql@numbering@tab@center
949 \let\eql@numbering@tab@m\eql@numbering@tab@median
950 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
951 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
952 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
953 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

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

```

TODO: describe

```
955 \def\eql@numbering@init@sub{%
956   \let\eql@numbering@multi\eql@true
957   \ifdefined\eql@subequations@active
958     \let\eql@numbering@mode\eql@numbering@tab@all
959   \else
960     \let\eql@numbering@subeq@use\eql@true
961   \fi
962 }

963 \def\eql@numbering@init@first{\eql@tagpos@row@\@ne}
964 \def\eql@numbering@init@last{\eql@tagpos@row@\@MM}
965 \def\eql@numbering@init@here{\eql@tagpos@row@\m@ne}
```

TODO: describe

```
966 \def\eql@numbering@init@in{%
967   \ifdefined\eql@tagsleft
968     \eql@numbering@init@last
969   \else
970     \eql@numbering@init@first
971   \fi
972 }
```

TODO: describe

```
973 \def\eql@numbering@init@out{%
974   \ifdefined\eql@tagsleft
975     \eql@numbering@init@first
976   \else
977     \eql@numbering@init@last
978   \fi
979 }
```

TODO: describe

```
980 \def\eql@tagpos@eval@middle{%
981   \ifnum\eql@tagpos@row@=\z@
982     \eql@tagpos@row@\numexpr(\eql@totalrows@
983       +\ifdefined\eql@tagsleft\z@\else\@ne\fi)/\tw@\relax
984   \fi
985 }
```

TODO: describe

```
986 \def\eql@tagpos@eval@best{%
987   \ifnum\eql@tagpos@row@=\z@
988     \let\eql@numbering@best@use\eql@true
989     \eql@numbering@init@out
990   \fi
991 }
```

TODO: describe

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

TODO: describe

```
993 \let\eql@numbering@init@top\eql@numbering@init@continuous
994 \def\eql@tagpos@eval@top{%
995   \eql@tagpos@current@\z@
996 }
```


TODO: describe

```
997 \let\eql@numbering@init@bottom\eql@numbering@init@continuous
998 \def\eql@tagpos@eval@bottom{%
999   \eql@tagpos@current@\dimexpr\eql@totalheight@
1000     -\eql@tagheight@block@-\eql@tagdepth@block@\relax
1001 }
```

TODO: describe

```
1002 \let\eql@numbering@init@center\eql@numbering@init@continuous
1003 \def\eql@tagpos@eval@center{%
1004   \ifnum\eql@totalrows@=\@ne
1005     \eql@tagpos@row@\@ne
1006   \fi
1007   \eql@tagpos@current@\dimexpr(\eql@totalheight@
1008     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
1009 }
```

TODO: describe

```
1010 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
1011 \def\eql@tagpos@eval@centerone{%
1012   \eql@tagpos@current@\dimexpr(\eql@totalheight@
1013     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
1014 }
```

TODO: describe

```
1015 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
1016 \def\eql@tagpos@eval@baseline{%
1017   \eql@tagpos@current@\dimexpr(\eql@totalheight@
1018     +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
1019 }
```

TODO: describe

```
1020 \let\eql@numbering@init@median\eql@numbering@init@continuous
1021 \def\eql@tagpos@eval@median{%
1022   \ifnum\eql@tagpos@row@=\z@
1023     \ifodd\eql@totalrows@
1024       \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
1025     \else
1026       \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
1027       \eql@dimensions@get\eql@tagpos@row@
1028       \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
1029         +(\eql@line@interline@-\eql@tagheight@block@
1030         +\eql@tagdepth@block@)/\tw@\relax
1031     \fi
1032   \ifnum\eql@totalrows@=\@ne
1033     \eql@tagpos@row@\@ne
1034   \else
1035     \eql@tagpos@adjust@eval@convert
1036     \eql@tagpos@row@\z@
1037   \fi
1038 \fi
1039 }
```

Selection.

```
1040 \def\eql@numbering@set#1{%
```

```

1041 \ifcsname eql@numbering@tab@#1\endcsname
1042   \expandafter\let\expandafter\eql@numbering@mode
1043   \csname eql@numbering@tab@#1\endcsname
1044   \ifx\eql@numbering@mode\eql@numbering@tab@all
1045     \let\eql@numbering@mode@multi\eql@numbering@mode
1046   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
1047     \let\eql@numbering@mode@multi\eql@numbering@mode
1048   \else
1049     \let\eql@numbering@mode@single\eql@numbering@mode
1050   \fi\fi
1051 \else
1052   \eqLError{numbering mode '#1' unknown: setting mode to 'all'}%
1053   \let\eql@numbering@mode\eql@numbering@tab@all
1054 \fi
1055 }

```

TODO: describe

```

1056 \def\eql@numbering@init{%
1057   \let\eql@numbering@multi\eql@false
1058   \let\eql@tagpos@continuous\eql@false
1059   \let\eql@numbering@subeq@use\eql@false
1060   \let\eql@numbering@best@use\eql@false
1061   \eql@tagpos@row@z@
1062   \csname eql@numbering@init@\eql@numbering@mode\endcsname
1063   \ifdefined\eql@numbering@active
1064     \let\eql@numbering@eqnswinit\@eqnswtrue
1065   \else
1066     \let\eql@numbering@eqnswinit\@eqnswfalse
1067   \fi
1068   \let\eql@numbering@active\eql@false
1069 }

```

5.3 Interface

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

```

1070 \eql@amsmath@after{
1071   \let\eql@print@eqnum@default\print@eqnum
1072   \let\eql@incr@eqnum@default\incr@eqnum
1073 }

```

TODO: describe

```

1074 \protected\def\donumber{%
1075   \if@eqnsw\else
1076     \global\@eqnswtrue
1077     \ifx\print@eqn\@empty
1078       \global\let\print@eqn\eql@print@eqnum@default
1079     \fi
1080     \ifx\incr@eqn\@empty
1081       \global\let\incr@eqn\eql@incr@eqnum@default
1082     \fi
1083   \fi
1084 }

```

TODO: reconsider operation

`\numberhere`

```

1085 \protected\def\eq@numberhere{%
1086   \ifdefined\eq@numbering@multi
1087     \global\@eqnswtrue
1088   \else
1089     \global\eq@tagpos@row@\eq@row@
1090   \fi
1091 }

```

TODO: describe

\numbernext

```

1092 \protected\def\eq@numbernext{%
1093   \ifdefined\eq@numbering@multi
1094     \global\@eqnswfalse
1095   \else
1096     \ifdefined\eq@tagpos@continuous\else
1097       \ifnum\eq@tagpos@row@=\eq@row@
1098         \global\advance\eq@tagpos@row@\@ne
1099       \fi
1100     \fi
1101   \fi
1102 }

```

Activation Trigger.

```

1103 \def\eq@tags@autoenable{%
1104   \global\@eqnswtrue
1105   \ifnum\eq@tagpos@row@=\m@ne
1106     \numberhere
1107   \fi
1108 }

```

Labels. **TODO:** describe

\eq@label@org

```

1109 \let\eq@label@org\label

```

TODO: describe

```

1110 \def\eq@label@gobble{\eq@ampprotect\eq@testopt@tight\eq@gobbleoptone{}}

```

TODO: describe

```

1111 \protected\def\eq@label{%
1112   \eq@ampprotect\eq@testopt@tight\eq@tags@add@labelname\eq@testopt@default
1113 }

```

TODO: describe

```

1114 \def\eq@tags@add@labelname[#1]#2{%
1115   \def\eq@tmp{#1}%
1116   \ifx\eq@tmp\eq@testopt@default\else
1117     \eq@tags@add@name{#1}%
1118   \fi
1119   \eq@tags@add@label{#2}%
1120 }

```

TODO: describe

```
1121 \def\eql@tags@set@label#1{%
1122   \ifdefined\eql@tags@warn
1123     \ifdefined\eql@tags@label
1124       \eql@warn@label@multiple{#1}%
1125     \fi
1126   \fi
1127   \def\eql@tags@label{#1}%
1128 }
```

TODO: describe

```
1129 \def\eql@tags@set@name#1{%
1130   \ifdefined\eql@tags@warn
1131     \ifdefined\eql@tags@name
1132       \eql@warn@name@multiple
1133     \fi
1134   \fi
1135   \def\eql@tags@name{#1}%
1136 }
```

TODO: describe

```
1137 \def\eql@tags@add@label#1{%
1138   \ifdefined\eql@tags@autolabel
1139     \eql@tags@autoenable
1140   \fi
1141   \global\eql@appendexpand\eql@tags@container{%
1142     \noexpand\eql@tags@set@label{#1}}%
1143 }
```

TODO: describe

```
1144 \def\eql@tags@add@name#1{%
1145   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1146   \global\eql@appendmacro\eql@tags@container\eql@tmp
1147 }
```

TODO: describe

```
1148 \def\eql@tags@addblock@label#1{%
1149   \eql@appendexpand\eql@tags@container@block{%
1150     \noexpand\eql@tags@set@label{#1}}%
1151 }
```

TODO: describe

```
1152 \def\eql@tags@addblock@name#1{%
1153   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1154   \eql@appendmacro\eql@tags@container@block\eql@tmp
1155 }
```

Tags. **TODO:** describe

`\eql@tag@default`

```
1156 \protected\def\eql@tag@default{%
1157   \eql@warn@here\tag
1158   \eql@tag@gobble
1159 }
1160 \let\tag\eql@tag@default
```

\eql@tag@gobble

```
1161 \def\eql@tag@gobble{%
1162   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}
```

TODO: describe

```
1163 \protected\def\eql@tag{%
1164   \eql@ampprotecttwo\eql@teststaropt@tight
1165   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
1166   \eql@testopt@default
1167 }
```

\eql@tags@add@tagref

```
1168 \def\eql@tags@add@tagref[#1]#2{%
1169   \def\eql@tmp{#1}%
1170   \ifx\eql@tmp\eql@testopt@default\else
1171     \eql@tags@add@ref{#1}%
1172   \fi
1173   \eql@tags@add@tag{#2}%
1174 }
```

TODO: describe

```
1175 \def\eql@tags@set@tag#1{%
1176   \ifdefined\eql@tags@warn
1177     \ifdefined\eql@tags@tag
1178       \eql@warn@tag@multiple
1179     \fi
1180   \fi
1181   \def\eql@tags@tag{#1}%
1182 }
```

TODO: describe

```
1183 \def\eql@tags@set@ref#1{%
1184   \ifdefined\eql@tags@warn
1185     \ifdefined\eql@tags@ref
1186       \eql@warn@ref@multiple
1187     \fi
1188   \fi
1189   \def\eql@tags@ref{#1}%
1190 }
```

TODO: describe

```
1191 \def\eql@tags@add@tag#1{%
1192   \ifdefined\eql@tags@autotag
1193     \eql@tags@autoenable
1194   \fi
1195   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
1196   \global\eql@appendmacro\eql@tags@container\eql@tmp
1197 }
```

TODO: describe

```
1198 \def\eql@tags@add@ref#1{%
1199   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
1200   \global\eql@appendmacro\eql@tags@container\eql@tmp
1201 }
```

tags@add@tagform@off

```
1202 \def\eq\@tags@add@tagform@off{%
1203   \global\eq\@append\eq\@tags@container{\let\eq\@tags@tagform\@firstofone}%
1204 }
```

TODO: describe

```
1205 \def\eq\@tags@addblock@tag#1{%
1206   \protected@edef\eq\@tmp{\noexpand\eq\@tags@set@tag{#1}}%
1207   \eq\@appendmacro\eq\@tags@container@block\eq\@tmp
1208 }
```

TODO: describe

```
1209 \def\eq\@tags@addblock@ref#1{%
1210   \protected@edef\eq\@tmp{\noexpand\eq\@tags@set@ref{#1}}%
1211   \eq\@appendmacro\eq\@tags@container@block\eq\@tmp
1212 }
```

TODO: describe

```
1213 \def\eq\@tags@addblock@tagform@off{%
1214   \eq\@append\eq\@tags@container@block{\let\eq\@tags@tagform\@firstofone}%
1215 }
```

Raise Tags.

\raisetag

```
1216 \def\eq\@raisetag@default{%
1217   \eq\@warn@here\raisetag
1218   \eq\@raisetag@gobble
1219 }

1220 \def\eq\@raisetag@gobble{%
1221   \eq\@ampprotecttwo\eq\@ifstar@tight\@gobble\@gobble
1222 }
```

TODO: describe

```
1223 \eq\@amsmath\let\raisetag\eq\@raisetag@default

1224 \def\eq\@raisetag{%
1225   \eq\@ampprotecttwo\eq\@ifstar@tight\eq\@tags@add@raiseshift\eq\@raisetag@test
1226 }

1227 \def\eq\@raisetag@test#1{%
1228   \def\eq\@tmpa{#1}\def\eq\@tmpb{!}%
1229   \ifx\eq\@tmpa\eq\@tmpb
1230     \eq\@tags@add@forceraise
1231   \else
1232     \eq\@tags@add@raisesmash{#1}%
1233   \fi
1234 }

1235 \def\eq\@tags@add@raiseshift#1{%
1236   \global\eq\@appendexpand\eq\@tags@container{%
1237     \advance\eq\@tagpos@shift@the\glueexpr#1\relax\relax}%
1238 }
```

```

1239 \def\eql@tags@add@raisesmash#1{%
1240   \dimen@{\glueexpr#1\relax
1241   \ifdim\dimen@<\z@
1242     \global\eql@appendexpand\eql@tags@container{%
1243       \advance\eql@tagpos@smashdown@-\the\dimen@\relax}%
1244   \else
1245     \global\eql@appendexpand\eql@tags@container{%
1246       \advance\eql@tagpos@smashup@\the\dimen@\relax}%
1247   \fi
1248 }

1249 \def\eql@tags@add@forceraise{%
1250   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
1251 }

```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```

1252 \def\eql@numbering@settools{%
1253   \let\label\eql@label
1254   \let\tag\eql@tag
1255   \let\raisetag\eql@raisetag
1256   \let\numberhere\eql@numberhere
1257   \let\numbernext\eql@numbernext
1258 }

```

TODO: not necessary anymore

```

1259 \def\eql@numbering@settools@gobble{%
1260   \let\label\eql@label@gobble
1261   \let\tag\eql@tag@gobble
1262   \let\raisetag\eql@raisetag@gobble
1263   \let\numberhere\relax
1264   \let\numbernext\relax
1265 }

```

```

1266 \def\eql@numbering@autoblock{%
1267   \begingroup
1268     \let\eql@tags@warn\eql@false
1269     \eql@tags@container@block
1270     \ifdefined\eql@tags@autolabel
1271       \ifdefined\eql@tags@label
1272         \global\@eqnswtrue
1273       \fi
1274     \fi
1275     \ifdefined\eql@tags@autotag
1276       \ifdefined\eql@tags@tag
1277         \global\@eqnswtrue
1278       \fi
1279     \fi
1280   \endgroup
1281 }

```

```

1282 \def\eql@numbering@warnunused{%
1283   \ifdefined\eql@tags@label

```

```

1284     \eql@warn@label@unused
1285 \fi
1286 \ifdefined\eql@tags@name
1287     \eql@warn@name@unused
1288 \fi
1289 \ifdefined\eql@tags@tag
1290     \eql@warn@tag@unused
1291 \fi
1292 \ifdefined\eql@tags@erf
1293     \eql@warn@ref@unused
1294 \fi
1295 }

```

Single Line. **TODO:** describe

```

1296 \def\eql@numbering@single@init{%
1297     \let\eql@numbering@multi\eql@false
1298     \eql@numbering@settools
1299     \eql@numbering@eqnswinit
1300     \eql@numbering@autoblock
1301     \global\let\eql@tags@container\eql@tags@container@block
1302     \let\eql@tags@warn\eql@true
1303 }

1304 \def\eql@numbering@single@eval{%
1305     \ifnum\eql@tagpos@row@=\m@ne
1306         \@eqnswfalse
1307     \fi
1308 }

```

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

```

1309 \def\eql@numbering@measure@init{%
1310     \eql@numbering@settools
1311     \ifdefined\eql@numbering@multi\else
1312         \eql@numbering@eqnswinit
1313         \eql@numbering@autoblock
1314     \fi
1315     \global\let\eql@tags@container\eql@tags@container@block
1316     \let\eql@tags@warn\eql@true
1317 }

```

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

```

1318 \def\eql@numbering@measure@line@begin{%
1319     \ifdefined\eql@numbering@multi
1320         \global\eql@numbering@eqnswinit
1321     \fi
1322 }

```

TODO: describe

```

1323 \def\eql@numbering@measure@blocktag{%
1324     \ifdefined\eql@numbering@multi
1325         \@eqnswfalse
1326     \else
1327         \ifnum\eql@tagpos@row@=\m@ne
1328             \@eqnswfalse
1329         \fi

```



```

1330     \ifnum\eql@totalrows@=\z@
1331         \eqnswfalse
1332     \fi
1333 \fi
1334 }

```

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!

```

1335 \def\eql@numbering@print@init{%
1336     \let\eql@tags@warn\eql@false
1337     \ifdefined\eql@numbering@multi
1338         \eql@numbering@settools
1339         \global\let\eql@tags@container\eql@tags@container@block
1340     \else
1341         \let\eql@tags@container@block\eql@tags@container
1342         \eql@numbering@settools@gobble
1343     \fi
1344 }

```

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

```

1345 \def\eql@numbering@print@block@begin{%
1346     \ifdefined\eql@numbering@multi\else
1347         \ifnum\eql@tagpos@row@>\z@
1348             \eql@tags@makeblockanchor
1349             \global\eql@appendexpand\eql@tags@container@block{%
1350                 \def\noexpand\eql@tags@anchor{%
1351                     \unexpanded\expandafter{\eql@tags@anchor}}}%
1352             \fi
1353         \fi
1354         \ifdefined\eql@numbering@subeq@use
1355             \eql@tags@printsubeqlabel
1356         \fi
1357 }

```

TODO: describe

```

1358 \def\eql@numbering@print@line@begin{%
1359     \ifdefined\eql@numbering@multi
1360         \global\eql@numbering@eqnswinit
1361     \fi
1362 }

```

TODO: describe

```

1363 \def\eql@numbering@print@line@eval{%
1364     \ifdefined\eql@numbering@multi
1365         \ifeqnsw
1366             \eql@tags@container
1367         \fi
1368     \else
1369         \ifnum\eql@tagpos@row@=\eql@row@
1370             \eqnswtrue
1371             \eql@tags@container@block
1372         \else
1373             \eqnswfalse
1374         \fi

```

```

1375 \fi
1376 }

```

5.5 Positioning

TODO: describe

```

1377 \def\eql@tagpos@single@eval{%
1378   \if@eqnsw
1379     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1380     \ifnum\eql@tagpos@row@>\@ne
1381       \eql@tagpos@row@\@ne
1382     \fi
1383     \ifdefined\eql@tagpos@doconvert
1384       \let\eql@tagpos@continuous\eql@true
1385     \fi
1386     \ifdefined\eql@tagpos@continuous
1387       \eql@tagpos@single@eval@continuous
1388     \fi
1389   \else
1390     \eql@tagpos@row@\z@
1391   \fi
1392   \eql@tagpos@prevrow@\z@
1393   \eql@tagpos@headroom@\z@
1394   \eql@tagpos@footroom@\z@
1395 }

```

TODO: describe

```

1396 \def\eql@tagpos@single@eval@continuous{%
1397   \ifnum\eql@tagpos@row@>\z@
1398     \eql@tagpos@target@\eql@tagpos@shift@
1399   \else
1400     \eql@tagpos@target@\dimexpr\eql@line@height@
1401       -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1402   \fi
1403   \eql@tagpos@row@\@ne
1404   \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1405     \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1406     \eql@tagpos@target@\z@
1407   \fi
1408 }

```

TODO: describe

```

1409 \def\eql@tagpos@adjust@eval{%
1410   \if@eqnsw
1411     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1412     \ifnum\eql@tagpos@row@>\eql@totalrows@
1413       \eql@tagpos@row@\eql@totalrows@
1414     \fi
1415     \ifdefined\eql@tagpos@doconvert
1416       \let\eql@tagpos@continuous\eql@true
1417     \fi
1418     \ifdefined\eql@tagpos@continuous
1419       \ifnum\eql@tagpos@row@>\z@
1420         \eql@tagpos@adjust@eval@convert
1421       \fi
1422       \eql@tagpos@adjust@eval@continuous

```

```

1423 \fi
1424 \else
1425 \eq@tagpos@row@z@
1426 \eq@tagpos@prevrow@z@
1427 \fi
1428 }

```

TODO: describe

```

1429 \def\eq@tagpos@adjust@eval@convert{%
1430 \eq@tagpos@current@z@
1431 \eq@dimensions@for{%
1432 \ifnum\eq@row@<\eq@tagpos@row@
1433 \advance\eq@tagpos@current@\dimexpr\eq@line@interline@
1434 +\eq@line@height@+\eq@line@depth@\relax
1435 \fi
1436 \ifnum\eq@row@=\eq@tagpos@row@
1437 \advance\eq@tagpos@current@\dimexpr\eq@line@interline@
1438 +\eq@line@height@-\eq@tagheight@block@\relax
1439 \fi
1440 }%
1441 }

```

TODO: describe

```

1442 \def\eq@tagpos@adjust@eval@continuous{%
1443 \dimen@\dimexpr\eq@tagpos@current@-\eq@tagpos@shift@\relax
1444 \eq@tagpos@row@\eq@totalrows@
1445 \eq@tagpos@prevrow@z@
1446 \eq@tagpos@headroom@z@
1447 \eq@tagpos@footroom@z@
1448 \eq@dimensions@for{%
1449 \ifnum\eq@tagpos@row@=\eq@totalrows@
1450 \eq@tagpos@headroom@\eq@line@interline@
1451 \eq@tagpos@target@\dimexpr\eq@line@interline@
1452 +\eq@line@height@-\dimen@-\eq@tagheight@block@\relax
1453 \ifdim\ifdim\eq@tagpos@target@<z@-\fi
1454 \eq@tagpos@target@<\glueexpr\eq@tagpos@snap\relax
1455 \advance\dimen@\eq@tagpos@target@
1456 \eq@tagpos@target@z@
1457 \fi
1458 \ifdim\eq@tagpos@target@>%
1459 \ifdefined\eq@tagsleft-1sp\relax\else\z@\fi
1460 \eq@tagpos@row@\eq@row@
1461 \eq@tagpos@prevrow@\numexpr\eq@row@-\@ne\relax
1462 \fi
1463 \advance\dimen@-\dimexpr\eq@line@interline@
1464 +\eq@line@depth@+\eq@line@height@\relax
1465 \fi
1466 \ifnum\eq@row@=\numexpr\eq@tagpos@row@+\@ne\relax
1467 \eq@tagpos@footroom@\eq@line@interline@
1468 \fi
1469 }%
1470 }

```

TODO: describe

```

1471 \def\eq@tagpos@print@line@eval{%
1472 \ifdefined\eq@tagpos@continuous
1473 \eq@tagpos@print@line@eval@continuous
1474 \else

```

```

1475 \eq\tagpos@print@line@eval@discrete
1476 \fi
1477 }

```

TODO: describe

```

1478 \def\eq\tagpos@print@line@eval@continuous{%
1479 \if@eqnsw
1480 \ht\eq\tagbox@\dimexpr\ht\eq\tagbox@-\eq\tagpos@smashup@\relax
1481 \dp\eq\tagbox@\dimexpr\dp\eq\tagbox@-\eq\tagpos@smashdown@\relax
1482 \eq\tagpos@plain@\eq\tagpos@target@
1483 \@tempdima\dimexpr\eq\line@height@+\eq\tagpos@headroom@
1484 -\ht\eq\tagbox@\relax
1485 \@tempdimb\dimexpr-\eq\line@depth@-\eq\tagpos@footroom@
1486 +\dp\eq\tagbox@\relax
1487 \ifnum\eq\row@=\@ne
1488 \@tempdima.5\maxdimen
1489 \fi
1490 \ifnum\eq\row@=\eq\totalrows@
1491 \@tempdimb-.5\maxdimen
1492 \fi
1493 \ifdim\eq\tagpos@plain@>\@tempdima
1494 \ifdim\eq\tagpos@plain@>\@tempdimb
1495 \ifdim\@tempdima>\@tempdimb
1496 \eq\tagpos@plain@\@tempdima
1497 \else
1498 \eq\tagpos@plain@\@tempdimb
1499 \fi
1500 \fi
1501 \else
1502 \ifdim\eq\tagpos@plain@<\@tempdimb
1503 \ifdim\@tempdima>\@tempdimb
1504 \eq\tagpos@plain@\@tempdimb
1505 \else
1506 \eq\tagpos@plain@\@tempdima
1507 \fi
1508 \fi
1509 \fi
1510 \ifnum\eq\tagpos@prevrow@>\z@
1511 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1512 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1513 \eq\tagpos@raised@\eq\tagpos@plain@
1514 \let\eq\tagpos@reserve\eq>false
1515 \fi
1516 \else
1517 \ifdim\eq\tagpos@target@>%
1518 \ifdefined\eq\tagleft-isp\relax\else\z@\fi
1519 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1520 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1521 \eq\tagpos@raised@\eq\tagpos@plain@
1522 \let\eq\tagpos@reserve\eq>false
1523 \fi
1524 \else
1525 \eq\tagpos@raised@\dimexpr-\eq\line@depth@
1526 -\ht\eq\tagbox@\relax
1527 \ifdim\eq\tagpos@raised@<\eq\tagpos@plain@\else
1528 \eq\tagpos@raised@\eq\tagpos@plain@
1529 \let\eq\tagpos@reserve\eq>false
1530 \fi

```

```

1531     \fi
1532   \fi
1533 \else
1534   \ifnum\eql@tagpos@prevrow@=\eql@row@
1535     \eql@tagwidth@\eql@tagwidth@block@
1536   \else
1537     \let\eql@tagpos@reserve\eql@false
1538   \fi
1539 \fi
1540 }

```

TODO: describe

```

1541 \def\eql@tagpos@print@line@eval@discrete{%
1542   \if@eqnsw
1543     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1544     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1545     \eql@tagpos@plain@\eql@tagpos@shift@
1546     \eql@tagpos@headroom@z@
1547     \eql@tagpos@footroom@z@
1548     \ifdim\eql@tagpos@shift@>%
1549       \ifdefined\eql@tagsleft-1sp\relax\else z@\fi
1550     \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1551   \else
1552     \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1553   \fi
1554 \else
1555   \let\eql@tagpos@reserve\eql@false
1556 \fi
1557 }

```

TODO: describe

```

1558 \def\eql@tagpos@print@line@end{%
1559   \ifdefined\eql@tagpos@continuous
1560     \ifnum\eql@tagpos@prevrow@=\eql@row@
1561       \ifdefined\eql@tagpos@reserve
1562         \global\eql@appendexpand\eql@tags@container@block{%
1563           \advance\eql@tagpos@headroom@the\dimexpr\eql@line@height@
1564             +\eql@line@depth@\relax\relax}%
1565         \eql@displaybreak@star\@M
1566       \fi
1567     \fi
1568   \fi
1569 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1570 \let\eql@SK@loaded\eql@false
1571 \let\eql@SK@label\@gobble
1572 \let\eql@SK@clearlabel\@empty
1573 \let\eql@SK@setlabel\@gobble
1574 \let\eql@SK@printlabel@right\@empty
1575 \let\eql@SK@printlabel@left\@empty
1576 \let\eql@SK@printlabel@line\@empty
1577 \def\eql@label@clean{\eql@label@org}
1578 \AddToHook{package/showkeys/after}{

```

```

1579 \let\eql@SK@loaded\eql@true
1580 \def\eql@SK@label#1{\SK@\SK@@label#1}
1581 \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1582 \eql@SK@clearlabel
1583 \def\eql@SK@@label#1>#2\SK@{%
1584   \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1585 }
1586 \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1587 \def\eql@SK@printlabel@right{%
1588   \ifx\eql@SK@lab\relax\else
1589     \rlap{\kern\marginparsep\eql@SK@lab}%
1590     \eql@SK@clearlabel
1591   \fi
1592 }
1593 \def\eql@SK@printlabel@left{%
1594   \ifx\eql@SK@lab\relax\else
1595     \llap{\eql@SK@lab\kern\marginparsep}%
1596     \eql@SK@clearlabel
1597   \fi
1598 }
1599 \def\eql@SK@printlabel@line{%
1600   \ifx\eql@SK@lab\relax\else
1601     \dimen@ \prevdepth
1602     \nointerlineskip
1603     \ifdefined\eql@tagsleft
1604       \llap{%
1605         \eql@SK@lab
1606         \kern\marginparsep
1607       }%
1608       \eql@SK@clearlabel
1609     \else
1610       \rlap{%
1611         \dimen@ \displaywidth
1612         \advance\dimen@ \marginparsep
1613         \kern\dimen@
1614         \eql@SK@lab
1615       }%
1616       \fi
1617       \eql@SK@clearlabel
1618       \prevdepth\dimen@
1619     \fi
1620 }
1621 \let\eql@label@org\label
1622 \def\eql@label@clean{\let\SK@\gobbletwo\eql@label@org}
1623 }

```

Labels.

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

```

1624 \def\eql@composetag@label{%
1625   \eql@SK@clearlabel
1626   \ifdefined\eql@tags@label
1627     \eql@SK@setlabel\eql@tags@label
1628     \ifdefined\eql@tags@name
1629       \let\@currentlabelname\eql@tags@name
1630     \else
1631       \let\@currentlabelname\eql@tags@name@generic

```

```

1632 \fi
1633 \expandafter\eq1@label@clean\expandafter{\eq1@tags@label}%
1634 \fi
1635 }

```

TODO: describe

```

1636 \def\eq1@tags@printsubeqlabel{%
1637 \eq1@tags@container@parent
1638 \ifdefined\eq1@tags@label
1639 \eq1@tags@makeblockanchor
1640 \eq1@SK@setlabel\eq1@tags@label
1641 \begingroup
1642 \def\@currentcounter{equation}%
1643 \eq1@tags@anchor
1644 \let\@currentlabelname\eq1@tags@name@generic
1645 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1646 \expandafter\eq1@label@clean\expandafter{\eq1@tags@label}%
1647 \endgroup
1648 \eq1@SK@printlabel@line
1649 \fi
1650 }

```

Hyperref Anchors. **TODO:** describe

```

1651 \let\eq1@Hy@anchor\@gobble
1652 \AddToHook{package/hyperref/after}{
1653 \def\eq1@Hy@anchor#1{%
1654 \Hy@raisedlink{\hyper@anchor{#1}}%
1655 }%
1656 }

```

TODO: describe

```

1657 \def\eq1@tags@makeblockanchor{%
1658 \eq1@tags@glabel@step
1659 \eq1@Hy@anchor\eq1@tags@glabel
1660 \edef\eq1@tags@anchor{%
1661 \def\noexpand\thepage{\thepage}%
1662 \def\noexpand\@currentHref{\eq1@tags@glabel}%
1663 }%
1664 }

```

TODO: describe

eq1@composetag@anchor

```

1665 \def\eq1@composetag@anchor{%
1666 \ifdefined\eq1@tags@tag
1667 \def\@currentcounter{equation}%
1668 \ifdefined\eq1@tags@ref
1669 \let\@currentlabel\eq1@tags@ref
1670 \else
1671 \protected@edef\@currentlabel{\p@equation\eq1@tags@tag}%
1672 \fi
1673 \eq1@tags@glabel@step
1674 \edef\@currentHref{\eq1@tags@glabel}%
1675 \eq1@Hy@anchor\@currentHref
1676 \else

```

```

1677 \refstepcounter{equation}%
1678 \protected@edef\eql@tags@tag{\theequation}%
1679 \fi
1680 \eql@tags@anchor
1681 }

```

Tag Layout. **TODO:** describe

```

1682 \def\eql@tags@taglayout@set@direct#1{%
1683 \def\eql@tags@taglayout##1{#1}%
1684 }
1685 \def\eql@tags@taglayout@set#1{%
1686 \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1687 }

```

TODO: describe

```

1688 \def\eql@tags@tagform@set@direct#1{%
1689 \def\eql@tags@tagform##1{#1}%
1690 }
1691 \def\eql@tags@tagform@set#1#2#3{%
1692 \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1693 }

1694 \eql@tags@taglayout@set{#1}
1695 \eql@tags@tagform@set({#1})
1696 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1697 \protected\def\tagform{\eql@tags@tagform}
1698 \protected\def\tagbox{\eql@tags@taglayout}
1699 \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:

```

1700 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1701 \def\eql@composetag@tag{%
1702 \eql@tagging@tagbegin
1703 \eql@tags@frame@cmd{%
1704 \eql@tags@taglayout{%
1705 \eql@tags@tagform\eql@tags@tag
1706 \eql@tagging@tagsave
1707 }%
1708 }%
1709 \eql@tagging@tagend
1710 }

```

5.7 Tag Composition

TODO: describe

```

1711 \def\eql@composetag@measure{%
1712 \ifdefined\eql@tags@tag\else
1713 \stepcounter{equation}%
1714 \let\eql@tags@tag\theequation

```



```

1715 \fi
1716 \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1717 \ifdefined\eql@numbering@multi
1718   \global\let\eql@tags@container\eql@tags@container@clear
1719 \fi
1720 }

```

TODO: describe

```

1721 \def\eql@composetag@print{%
1722   \eql@composetag@anchor
1723   \eql@composetag@label
1724   \ifdefined\eql@tagsleft
1725     \eql@SK@printlabel@left
1726     \eql@composetag@tag
1727   \else
1728     \eql@composetag@tag
1729     \eql@SK@printlabel@right
1730   \fi
1731   \global\let\eql@tags@container\eql@tags@container@clear
1732 }

```

TODO: describe

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

```

1733 \def\eql@tagbox@make#1{%
1734   \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
1735   \eql@tagwidth@\wd\eql@tagbox@
1736   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1737     \eql@tagwidth@\eql@tagwidthmin@
1738   \fi
1739   \advance\eql@tagwidth@\eql@tagsepmin@
1740 }

```

TODO: describe

```

1741 \def\eql@tagbox@print@adjustheadroom{%
1742   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1743   \ifdim\dimen@>\z@
1744     \ifdim\dimen@>\eql@tagpos@headroom@
1745       \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1746     \else
1747       \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1748     \fi
1749   \fi
1750 }

```

TODO: describe

```

1751 \def\eql@tagbox@print@adjustfootroom{%
1752   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1753   \ifdim\dimen@>\z@
1754     \ifdim\dimen@>\eql@tagpos@footroom@
1755       \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1756     \else
1757       \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1758     \fi
1759   \fi
1760 }

```

TODO: describe

```

1761 \def\eq\tagbox@print@extendabove{%
1762   \dimen@ \dimexpr \ht\eq\tagbox@+\eq\tagpos@current@-\eq\line@height@ \relax
1763   \ifdim \dimen@ > \z@
1764     \global\eq\appendexpand\eq\display@container{%
1765       \eq\display@aboveextend@ \the\dimen@ \relax}%
1766   \fi
1767 }

```

TODO: describe

```

1768 \def\eq\tagbox@print@extendbelow{%
1769   \dimen@ \dimexpr \dp\eq\tagbox@-\eq\tagpos@current@-\eq\line@depth@ \relax
1770   \ifdim \dimen@ > \z@
1771     \global\eq\appendexpand\eq\display@container{%
1772       \eq\display@belowextend@ \the\dimen@ \relax}%
1773   \fi
1774 }

```

TODO: describe

```

1775 \def\eq\tagbox@print@prepare{%
1776   \ifdefined\eq\tagpos@reserve
1777     \eq\tagpos@current@ \eq\tagpos@plain@
1778   \else
1779     \eq\tagpos@current@ \eq\tagpos@raised@
1780   \fi
1781   \ifdim \eq\tagpos@headroom@ > \z@
1782     \eq\tagbox@print@adjustheadroom
1783   \fi
1784   \ifdim \eq\tagpos@footroom@ > \z@
1785     \eq\tagbox@print@adjustfootroom
1786   \fi
1787   \ifnum \eq\row@ = \one
1788     \eq\tagbox@print@extendabove
1789   \fi
1790   \ifnum \eq\row@ = \eq\totalrows@
1791     \eq\tagbox@print@extendbelow
1792   \fi
1793 }

```

TODO: describe

```

1794 \def\eq\tagbox@print@tagsright{%
1795   \eq\tagbox@print@prepare
1796   \kern-\wd\eq\tagbox@
1797   \raise\eq\tagpos@current@\box\eq\tagbox@
1798 }

```

TODO: describe

```

1799 \def\eq\tagbox@print@tagsleft{%
1800   \eq\display@firstavail@set\z@
1801   \eq\tagbox@print@prepare
1802   \wd\eq\tagbox@ \z@
1803   \raise\eq\tagpos@current@\box\eq\tagbox@
1804 }

```

$\tagbox@print@cell$

```

1805 \def\eq\tagbox@print@cell{%
1806   \eq\tagging@tagaddbox

```

```

1807 \ifdefined\eq\@tagsleft
1808 \ifdefined\eq\@tagpos@reserve
1809 \ifdim\eq\@tagwidth@>\dimexpr\eq\@line@avail@+\eq\@tagfuzz@\relax
1810 \let\eq\@tagpos@reserve\eq\@false
1811 \fi
1812 \fi
1813 \if@eqnsw
1814 \eq\@tagbox@print@tagsleft
1815 \fi
1816 \kern\displaywidth
1817 \else
1818 \kern\displaywidth
1819 \ifdefined\eq\@tagpos@reserve
1820 \ifdim\eq\@tagwidth@>%
1821 \dimexpr\displaywidth-\eq\@line@width@+\eq\@tagfuzz@\relax
1822 \let\eq\@tagpos@reserve\eq\@false
1823 \fi
1824 \fi
1825 \if@eqnsw
1826 \eq\@tagbox@print@tagsright
1827 \fi
1828 \fi
1829 }

```

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:

```

1830 \eq\@amsmath@undefine\c@parentequation
1831 \eq\@amsmath@undefine\theparentequation
1832 \ifdefined\c@parentequation\else
1833 \newcounter{parentequation}
1834 \fi

```

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

```

1835 \def\eq\@subequations@template{\theparentequation\alph{equation}}

```

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

```

1836 \let\eq\@subequations@active\eq\@false

```

`\eq\@subequations@init` Low-level initialise the subequations mode. Store the equation counter in `\eq\@subequations@restorecounter` for the case that no equation numbers will be used. Step the equation counter, copy it to `\parentequation` and initialise `\theparentequation` (and its `\hyperref` counterpart) with the expanded current value of `\theequation`; fill with

tag data instead if a tag has been specified. Reset the equation counter and use the template for `\theequation`:

```

1837 \def\eq@subequations@init{%
1838   \edef\eq@subequations@restorecounter{%
1839     \global\c@equation\the\c@equation\relax}%
1840   \eq@tags@container@block
1841   \ifdefined\eq@tags@tag
1842     \eq@tags@glabel@step
1843     \protected@edef\theHparentequation{\eq@tags@glabel}%
1844     \protected@edef\theparentequation{\eq@tags@tag}%
1845   \else
1846     \advance\c@equation\@ne
1847     \protected@edef\theparentequation{\theequation}%
1848     \ifdefined\theHequation
1849       \protected@edef\theHparentequation{\theHequation}%
1850     \fi
1851   \fi
1852   \global\c@parentequation\c@equation
1853   \global\c@equation\z@
1854   \let\theequation\eq@subequations@template
1855   \def\theHequation{\theHparentequation.\arabic{equation}}%
1856 }
```

`\@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:

```

1857 \def\eq@subequations@close{%
1858   \ifnum\c@equation=\z@
1859     \eq@subequations@restorecounter
1860   \else
1861     \global\c@equation\c@parentequation
1862   \fi
1863 }
```

6.2 Environment

`\@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@printsubeqlabel`

```

1864 \def\eq@subequations@start{%
1865   \let\eq@tags@container@block\eq@tags@container@clear
1866   \eq@nextopt@process{subequations}%
1867   \eq@subequations@init
1868   \eq@tags@glabel@step
1869   \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1870   \eq@Hy@anchor\eq@subequations@currentHref
1871   \edef\eq@subequations@thepage{\thepage}%
1872   \def\@currentcounter{equation}%
1873   \let\@currentHref\eq@subequations@currentHref
1874   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1875   \eq@tags@container@block
1876   \ifdefined\eq@tags@name
1877     \let\@currentlabelname\eq@tags@name
1878   \else
```

```

1879 \let\@currentlabelname\eq\@tags\@name\@generic
1880 \fi
1881 \let\eq\@subequations\@active\eq\@true
1882 \ifdefined\eq\@tags\@label
1883 \eq\@SK\@label\eq\@tags\@label
1884 \fi
1885 \ignorespaces
1886 }

```

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

```

1887 \def\eq\@subequations\@end{%
1888 \ifnum\c\@equation>\z@
1889 \eq\@tags\@container\@block
1890 \ifdefined\eq\@tags\@label
1891 \begingroup
1892 \def\@currentcounter{equation}%
1893 \let\thepage\eq\@subequations\@thepage
1894 \let\@currentHref\eq\@subequations\@currentHref
1895 % \TODO how about tag* ?! also within equations!
1896 \protected\edef\@currentlabel{\p\@equation\theparentequation}%
1897 \ifdefined\eq\@tags\@name
1898 \let\@currentlabelname\eq\@tags\@name
1899 \else
1900 \let\@currentlabelname\eq\@tags\@name\@generic
1901 \fi
1902 \expandafter\eq\@label\@clean\expandafter{\eq\@tags\@label}%
1903 \endgroup
1904 \fi
1905 \fi
1906 \eq\@subequations\@close
1907 }

```

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

```

1908 \newenvironment{eq\@subequations}{%
1909 \eq\@verbose\@info\eq\@verbose\@msg\@enterenv
1910 \eq\@subequations\@testall\eq\@subequations\@start%
1911 }{%
1912 \eq\@subequations\@end
1913 \ignorespacesafterend
1914 \eq\@verbose\@info\eq\@verbose\@msg\@leaveenv
1915 }

```

TODO: describe

```

1916 \def\eq\@subequations\@testall{\eq\@parseopt\@env\eq\@subequations\@testall\@parse}
1917 \def\eq\@subequations\@testall\@parse{%
1918 \ifx\eq\@parseopt\@token[%]
1919 \let\eq\@parseopt\@next\eq\@parseopt\@opt
1920 \fi
1921 \ifx\eq\@parseopt\@token\eq\@atxi
1922 \let\eq\@parseopt\@next\eq\@parseopt\@label
1923 \fi
1924 \ifx\eq\@parseopt\@token\eq\@atxii
1925 \let\eq\@parseopt\@next\eq\@parseopt\@label
1926 \fi

```

```

1927 \ifx\eql@parseopt@token\label
1928   \let\eql@parseopt@next\eql@parseopt@end
1929 \fi
1930 }

```

6.3 Subequation Scheme

TODO: describe

```

1931 \def\eql@numbering@subeq@init{%
1932   \let\eql@save@theequation\theequation
1933   \let\eql@save@theHequation\theHequation
1934   \eql@subequations@init
1935   \let\eql@tags@container@parent\eql@tags@container@block
1936   \let\eql@tags@container@block\eql@tags@container@clear
1937 }

```

TODO: describe

```

1938 \def\eql@numbering@subeq@test{%
1939   \ifnum\eql@tagrows@<\tw@
1940     \let\eql@tags@container@block\eql@tags@container@parent
1941     \let\eql@numbering@subeq@use\eql@false
1942     \let\theequation\eql@save@theequation
1943     \let\theHequation\eql@save@theHequation
1944     \eql@subequations@restorecounter
1945   \fi
1946 }

```

TODO: describe

```

1947 % \TODO note must not use setcounter here (when calc is loaded)
1948 \def\eql@numbering@subeq@close{%
1949   \eql@subequations@close
1950 }

```

7 Display Equations Support

TODO: describe

```

1951 \let\eql@display@injectbefore\@undefined
1952 \let\eql@display@injectafter\@undefined
1953 \let\eql@interline@container\@undefined
1954 \def\eql@interline@container@clear{%
1955   \eql@displaybreak@open@\@MM
1956   \eql@vspaceskip@\z@skip
1957 }

```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```

1958 \interdisplaylinepenalty\@M

```

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

```
1959 \def\eqldisplaybreak@open#1{%
1960   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1961 }
```

TODO: allow a displaybreak before equations

```
1962 \protected\def\eqldisplaybreak@default{%
1963   \eqldisplaybreak@warning{Invalid use of \string\displaybreak}{}%
1964   \eqldisplaybreak@teststaroropt@loose\@gobble\eqldisplaybreak@opt}%
1965 \eqldisplaybreak@after{\let\eqldisplaybreak@default\displaybreak}
1966 \eqldisplaybreak@let\displaybreak\eqldisplaybreak@default
```

```
1967 \newcount\eqldisplaybreak@pen@
1968 \newcount\eqldisplaybreak@prepen@
1969 \newcount\eqldisplaybreak@postpen@
```

TODO: describe

```
1970 \protected\def\eqldisplaybreak{%
1971   \relax
1972   \eqldisplaybreak@protecttwo\eqldisplaybreak@teststaroropt@tight
1973   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1974 }
```

```
1975 \def\eqldisplaybreak@star#1{%
1976   \global\eqldisplaybreak@appendexpand\eqldisplaybreak@interline@container{%
1977     \eqldisplaybreak@open@the\numexpr#1\relax\relax}%
1978 }
```

```
1979 \def\eqldisplaybreak@level[#1]{%
1980   \ifnum#1<\z@
1981     \global\eqldisplaybreak@append\eqldisplaybreak@interline@container{\eqldisplaybreak@open@\@MM}%
1982   \else
1983     \global\eqldisplaybreak@appendexpand\eqldisplaybreak@interline@container{%
1984       \eqldisplaybreak@open@-\@getpen{#1}\relax}%
1985   \fi
1986 }
```

TODO: describe

```
1987 \def\eqldisplaybreak@pre#1{%
1988   \ifnum#1<\z@
1989     \eqldisplaybreak@prepen@\@MM
1990   \else
1991     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1992   \fi
1993 }
```

TODO: describe

```
1994 \def\eqldisplaybreak@post#1{%
1995   \ifnum#1<\z@
1996     \eqldisplaybreak@postpen@\@MM
1997   \else
1998     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1999   \fi
2000 }
```

TODO: describe

```

2001 \def\eqldisplaybreak@inter#1{%
2002   \ifnum#1<\z@
2003     \interdisplaylinepenalty\@M
2004   \else
2005     \interdisplaylinepenalty\eqldgetdsp@pen{#1}\relax
2006   \fi
2007 }

```

7.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```

2008 \newskip\eqlvspaceskip@

2009 \let\eqlvspace@org\vspace
2010 \def\eqlvspace{%
2011   \ifvmode
2012     \expandafter\eqlvspace@immediate
2013   \else
2014     \expandafter\eqlvspace@line
2015   \fi
2016 }

```

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

```

2017 \def\eqlvspace@line{%
2018   \eqldifstar@loose\eqlvspace@addfixedbefore\eqlvspace@add
2019 }
2020 \def\eqlvspace@add#1{%
2021   \global\eqldappendexpand\eqldinterline@container{%
2022     \advance\eqlvspaceskip@\the\glueexpr#1\relax\relax}}
2023 \def\eqlvspace@addfixedbefore#1{%
2024   \global\eqldappendexpand\eqldinterline@container{%
2025     \noexpand\eqldappend\noexpand\eqldisplay@injectbefore{%
2026       \skip@\the\glueexpr#1\relax\relax
2027       \penalty\@M
2028       \vskip\skip@
2029       \global\advance\eqldline@interline@\skip@
2030     }%
2031   }%
2032 }
2033 \def\eqlvspace@addfixedafter#1{%
2034   \global\eqldappendexpand\eqldinterline@container{%
2035     \noexpand\eqldappend\noexpand\eqldisplay@injectafter{%
2036       \dimen@\prevdepth
2037       \hrule\@height\z@
2038       \skip@\the\glueexpr#1\relax\relax
2039       \penalty\@M
2040       \vskip\skip@
2041       \global\advance\eqldline@interline@\skip@
2042       \prevdepth\dimen@
2043     }%
2044   }%
2045 }

```


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

```

2046 \def\eqlvspace@immediate{%
2047   \noalign\bgroup
2048     \eq@ifstar@loose\eqlvspace@fixed\eqlvspace@discardable
2049 }
2050 \def\eqlvspace@fixed#1{%
2051   \skip@\glueexpr#1\relax
2052   \ifnum\eql@row@=\@ne
2053     \global\eql@appendexpand\eqldisplay@container{%
2054       \advance\eql@abovespace@\the\skip@\relax}%
2055   \else\ifnum\eql@row@>\eql@totalrows@
2056     \global\eql@appendexpand\eqldisplay@container{%
2057       \advance\eql@belowspace@\the\skip@\relax}%
2058   \else
2059     \dimen@\prevdepth
2060     \hrule\@height\z@
2061     \penalty\@M
2062     \vskip\skip@
2063     \global\advance\eql@line@interline@\skip@
2064     \prevdepth\dimen@
2065   \fi\fi
2066 \egroup
2067 }
2068 \def\eqlvspace@discardable#1{%
2069   \skip@\glueexpr#1\relax
2070   \ifnum\eql@row@=\@ne
2071     \global\eql@appendexpand\eqldisplay@container{%
2072       \advance\eql@abovespace@\the\skip@\relax}%
2073   \else\ifnum\eql@row@>\eql@totalrows@
2074     \global\eql@appendexpand\eqldisplay@container{%
2075       \advance\eql@belowspace@\the\skip@\relax}%
2076   \else
2077     \vskip\skip@
2078     \global\advance\eql@line@interline@\skip@
2079   \fi\fi
2080 \egroup
2081 }

```

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
`\eql@strutbox@` of a normal `\strut` minus `\normallineskiplimit` according to M. Spivak.

```

2082 \newbox\eql@strutbox@
2083 \def\eql@strut@depth{.3}
2084 \def\eql@strut{\copy\eql@strutbox@}
2085 \let\eql@strut@cell\eql@strut
2086 \let\eql@strut@tag\eql@strut
2087 \def\eql@strut@make{%
2088   \setbox\eql@strutbox@\hbox{%
2089     \@tempdimb\dimexpr
2090       \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
2091     \@tempdima\dimexpr
2092       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
2093     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@

```

```

2094 }
2095 }
2096 \AtBeginDocument{\eql@strut@make}

```

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

```

2097 \def\eql@spread@set{%
2098   \ifdefined\eql@spread@reset
2099     \lineskip\normallineskip
2100     \lineskiplimit\normallineskiplimit
2101     \baselineskip\normalbaselineskip
2102   \fi
2103   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
2104     +\normalbaselineskip-\baselineskip\relax
2105   \ifdim\eql@spread@>\z@
2106     \openup\eql@spread@
2107     \ifdefined\spread@equation
2108       \let\spread@equation\@empty
2109     \fi
2110   \fi
2111 }

```

7.4 Entry and Exit

TODO: describe

```

2112 \let\eql@beamerbasecolor@fix\@empty
2113 \AddToHook{package/beamerbasecolor/after}{%
2114   \def\eql@beamerbasecolor@fix{%
2115     \donotcolorouterdisplaymaths
2116     \donotcoloroutermaths
2117     \beamer@setdisplaymathcolor
2118   }%
2119 }

```

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

```

2120 \newskip\eql@abovespace@
2121 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

2122 \def\eql@display@enter{%
2123   \if@noskipsec\leavevmode\par\fi
2124   \ifvmode
2125     \eql@prevdepth@\prevdepth
2126     \nointerlineskip
2127     \noindent
2128   \else
2129     \eql@prevdepth@\maxdimen
2130   \fi
2131   \eql@beamerbasecolor@fix
2132 }

```

`\eql@display@adjust`

```

2133 \def\eql@display@adjust{%
2134   \ifdefined\eql@display@linewidth

```

```

2135 \displaywidth\glueexpr\eqldisplay@linewidth\relax
2136 \advance\displaywidth-\displayindent
2137 \fi
2138 \ifdefined\eqldisplay@marginleft
2139 \advance\displaywidth\displayindent
2140 \displayindent\glueexpr\eqldisplay@marginleft\relax
2141 \advance\displaywidth-\displayindent
2142 \fi
2143 \ifdefined\eqldisplay@marginright
2144 \advance\displaywidth-\glueexpr\eqldisplay@marginright\relax
2145 \fi
2146 \ifdim\displaywidth<\z@
2147 \displaywidth\z@
2148 \fi
2149 }

```

\eqldisplay@init

```

2150 \def\eqldisplay@init{%
2151 \let\eqldisplay@restore\eqldisplay@restore@active
2152 \let\displaybreak\eqldisplaybreak
2153 \let\eqlvspace@org\vspace
2154 \let\vspace\eqlvspace
2155 \let\eqncontrol\eql@control
2156 \let\eqldisplay@injectbefore\@empty
2157 \let\eqldisplay@injectafter\@empty
2158 \let\eqnpunct\eql@punct@setnext
2159 \eql@spread@set
2160 \eql@strut@make
2161 \let\eql@frame@cmd\@undefined
2162 }

```

\eqldisplay@print

```

2163 \def\eqldisplay@print{%
2164 \eql@punct@top@set
2165 \let\eqldisplay@container\@empty
2166 \eqldisplay@firstavail@\z@
2167 \eqldisplay@aboveextend@\z@
2168 \eqldisplay@belowextend@\z@
2169 \global\let\eql@interline@container\eql@interline@container@clear
2170 }

```

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

```

2171 \def\eqldisplay@halign@init#1{%
2172 \eql@row@\z@
2173 \eql@prevgraf@\prevgraf
2174 \everycr{\noalign{%
2175 \global\advance\eql@row@\@ne
2176 \prevgraf\numexpr\prevgraf+\@ne\relax
2177 #1%
2178 }}%
2179 }

```

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

```

2180 \def\eqldisplay@halign@start{%
2181 \prevgraf\numexpr\eql@prevgraf+\@ne\relax

```

```

2182 \ifdim\eql@prevdepth@=\maxdimen\else
2183   \prevdepth\eql@prevdepth@
2184 \fi
2185 \ifdim\prevdepth=-\@m\p@\else
2186   \ifdefined\eql@display@height
2187     \skip@ \baselineskip
2188     \advance\skip@-\glueexpr\eql@display@height\relax
2189     \advance\skip@-\prevdepth\relax
2190     \ifdim\skip@<\lineskiplimit
2191       \skip@ \lineskip
2192     \fi
2193     \advance\skip@-\eql@spread@\relax
2194     \vskip\skip@
2195     \nointerlineskip
2196   \else
2197     \vskip-\eql@spread@\relax
2198   \fi
2199 \fi
2200 }

```

TODO: describe

```

2201 \def\eql@display@vspace{%
2202   \advance\abovedisplayskip\eql@abovespace@
2203   \advance\belowdisplayskip\eql@belowspace@
2204 }

```

TODO: describe

```

2205 \def\eql@display@vspace@native{%
2206   \advance\abovedisplayskip\eql@abovespace@
2207   \advance\belowdisplayskip\eql@belowspace@
2208   \advance\abovedisplayshortskip\eql@abovespace@
2209   \advance\belowdisplayshortskip\eql@belowspace@
2210 }

```

TODO: describe

```

2211 \def\eql@display@penalty{%
2212   \ifnum\eql@displaybreak@postpen@=\@MM\else
2213     \postdisplaypenalty\eql@displaybreak@postpen@
2214   \fi
2215   \ifnum\eql@displaybreak@open@=\@MM\else
2216     \postdisplaypenalty\eql@displaybreak@open@
2217   \fi
2218   \ifnum\eql@displaybreak@prepen@=\@MM\else
2219     \predisplaypenalty\eql@displaybreak@prepen@
2220   \fi
2221 }

```

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

```

2222 \def\eql@display@halign@end{%
2223   \eql@interline@container
2224   \eql@display@injectbefore
2225   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
2226   \ifdefined\eql@display@depth
2227     \prevdepth\glueexpr\eql@display@depth\relax
2228   \fi
2229 }

```

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

```

2230 \def\eql@display@close{%
2231   \eql@display@container
2232   \ifdim\eql@display@firstavail@<\z@
2233     \eql@display@firstavail@>\z@
2234   \fi
2235   \eql@skip@mode@leave@>\z@
2236   \ifdim\eql@prevdepth@=\maxdimen
2237     \ifdim\predisplaysize=-\maxdimen
2238       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
2239       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
2240     \else
2241       \eql@skip@mode@above@>\z@
2242       \eql@skip@mode@below@>\z@
2243       \advance\eql@display@firstavail@>\displayindent
2244       \ifdim\eql@display@firstavail@>\predisplaysize
2245         \ifcase\eql@skip@mode@short\relax
2246         \or
2247           \eql@skip@mode@above@>\@ne
2248         \or
2249           \eql@skip@mode@above@>\@ne
2250           \ifnum\eql@totalrows@=\@ne
2251             \eql@skip@mode@below@>\@ne
2252           \fi
2253         \or
2254           \eql@skip@mode@above@>\@ne
2255           \eql@skip@mode@below@>\@ne
2256         \fi
2257       \fi
2258     \fi
2259   \else
2260     \ifdim\eql@prevdepth@=-\@m\p@
2261       \eql@skip@mode@above@\eql@skip@mode@top@above\relax
2262       \eql@skip@mode@below@\eql@skip@mode@top@below\relax
2263     \else
2264       \eql@skip@mode@above@\eql@skip@mode@par@above\relax
2265       \eql@skip@mode@below@\eql@skip@mode@par@below\relax
2266     \fi
2267   \fi
2268   \ifcase\eql@skip@mode@above@
2269   \or\or\or
2270     \predisplaypenalty\z@
2271   \or
2272     \predisplaypenalty\z@
2273   \fi
2274   \ifcase\eql@skip@mode@below@
2275   \or\or\or
2276     \eql@skip@mode@leave@>\@ne
2277   \or
2278     \eql@skip@mode@leave@>\tw@
2279   \fi
2280   \ifdefined\eql@skip@force@above
2281     \eql@skip@mode@above@\eql@skip@force@above\relax
2282   \fi
2283   \ifdefined\eql@skip@force@below
2284     \eql@skip@mode@below@\eql@skip@force@below\relax

```

```

2285 \fi
2286 \ifdefined\eq@skip@force@leave
2287 \eq@skip@mode@leave@\eq@skip@force@leave\relax
2288 \fi
2289 \ifnum\eq@skip@mode@leave@>\z@
2290 \postdisplaypenalty\z@
2291 \fi
2292 \ifcase\eq@skip@mode@above@
2293 \abovedisplayskip\glueexpr\eq@skip@long@above\relax
2294 \or
2295 \abovedisplayskip\glueexpr\eq@skip@short@above\relax
2296 \or
2297 \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
2298 \or
2299 \abovedisplayskip\glueexpr\eq@skip@par@above\relax
2300 \or
2301 \abovedisplayskip\glueexpr\eq@skip@top@above\relax
2302 \or
2303 \abovedisplayskip\z@skip
2304 \or
2305 \abovedisplayskip\glueexpr\eq@skip@med@above\relax
2306 \or
2307 \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
2308 \fi
2309 \ifcase\eq@skip@mode@below@
2310 \belowdisplayskip\glueexpr\eq@skip@long@below\relax
2311 \or
2312 \belowdisplayskip\glueexpr\eq@skip@short@below\relax
2313 \or
2314 \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
2315 \or
2316 \belowdisplayskip\glueexpr\eq@skip@par@below\relax
2317 \or
2318 \belowdisplayskip\glueexpr\eq@skip@top@below\relax
2319 \or
2320 \belowdisplayskip\z@skip
2321 \or
2322 \belowdisplayskip\glueexpr\eq@skip@med@below\relax
2323 \or
2324 \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
2325 \fi
2326 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
2327 \eq@interline@container
2328 \advance\eq@belowspace@\eq@vspaceskip@
2329 \eq@display@penalty
2330 \eq@display@vspace
2331 \skip@\glueexpr\eq@skip@tag@above\relax
2332 \ifdim\skip@>\abovedisplayskip
2333 \skip@\abovedisplayskip
2334 \fi
2335 \advance\abovedisplayskip-\eq@display@aboveextend@\relax
2336 \ifdim\abovedisplayskip<\skip@
2337 \abovedisplayskip\skip@
2338 \fi
2339 \skip@\glueexpr\eq@skip@tag@below\relax
2340 \ifdim\skip@>\belowdisplayskip
2341 \skip@\belowdisplayskip
2342 \fi

```

```

2343 \ifdim\eqldisplay@belowextend@>\z@
2344 \advance\belowdisplayskip-\eqldisplay@belowextend@\relax
2345 \ifdim\belowdisplayskip<\skip@
2346 \belowdisplayskip\skip@
2347 \fi
2348 \fi
2349 }

```

TODO: describe

```

2350 \def\eqldisplay@leave{%
2351 \prevgraf\eql@prevgraf@
2352 \ifcase\eql@skip@mode@leave@
2353 \or
2354 \endgraf
2355 \or
2356 \endgraf
2357 \prevdepth-\@m\p@
2358 \fi
2359 }

```

TODO: describe

```

2360 \def\eqldisplay@nest{%
2361 \let\displaybreak\eqldisplaybreak@default
2362 \let\intertext\eql@intertext@default
2363 \let\vspace\eql@vspace@org
2364 }

```

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

```

2365 \def\eqldisplay@restore@active{%
2366 \let\label\eql@label@org
2367 \let\tag\eql@tag@default
2368 \let\raisetag\eql@raisetag@default
2369 \let\displaybreak\eqldisplaybreak@default
2370 \let\intertext\eql@intertext@default
2371 \let\vspace\eql@vspace@org
2372 \ifdefined\eql@amp@mode
2373 \let\&\eql@amp@org
2374 \fi
2375 \let\eqnpunct\eql@punct@adopt
2376 \let\eql@punct@block\@undefined
2377 \let\eqldisplay@restore\@empty
2378 }

```

TODO: describe

```

2379 \let\eqldisplay@restore\@empty
2380 \eql@append\@arrayparboxrestore{%
2381 \eqldisplay@restore
2382 \ifdefined\eql@ampproof@active
2383 \eql@amp@prevert
2384 \fi
2385 \@displayfalse
2386 }

```

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)

```
2387 \def\eql@stack@enable{%
2388   \let\eql@stack@save@equations\eql@stack@save@equations@
2389   \let\eql@stack@save@box\eql@stack@save@box@
2390 }
```

TODO: describe

```
2391 \let\eql@stack@save@equations\eql@stack@enable
2392 \let\eql@stack@save@box\eql@stack@enable
2393 \let\eql@stack@restore\@empty
```

TODO: describe

```
2394 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
2395 \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

```
2396 \def\eql@stack@save@equations@{%
2397   \let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2398   \let\eql@stack@cell@container\eql@cell@container
2399   \let\eql@stack@tags@container\eql@tags@container
2400   \let\eql@stack@interline@container\eql@interline@container
2401   \let\eql@stack@dimensions@tab\eql@dimensions@tab
2402   \let\eql@stack@block@container\eql@display@container
2403   \let\eql@stack@punct@top\eql@punct@top
2404   \edef\eql@stack@restore{%
2405     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2406     \eql@stack@save@let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2407     \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2408     \eql@stack@save@let\eql@stack@tags@container\eql@tags@container
2409     \eql@stack@save@let\eql@stack@interline@container\eql@interline@container
2410     \eql@stack@save@let\eql@stack@dimensions@tab\eql@dimensions@tab
2411     \eql@stack@save@let\eql@stack@block@container\eql@display@container
2412     \eql@stack@save@let\eql@stack@punct@top\eql@punct@top
2413     \eql@stack@save@reg\eql@column@
2414     \eql@stack@save@reg\eql@totalcolumns@
2415     \eql@stack@save@reg\eql@line@avail@
2416     \eql@stack@save@reg\eql@line@pos@
2417     \eql@stack@save@reg\eql@line@width@
2418     \eql@stack@save@reg\eql@line@depth@
2419     \eql@stack@save@reg\eql@line@height@
2420     \eql@stack@save@reg\eql@line@prevdepth@
2421     \eql@stack@save@reg\eql@line@interline@
2422     \eql@stack@save@reg\eql@totalheight@
2423     \eql@stack@save@reg\eql@tagwidth@max@
2424     \eql@stack@save@reg\eql@tagpos@row@
2425     \eql@stack@save@reg\eql@row@
2426     \eql@stack@save@reg\eql@tagrows@
2427   }%
2428 }
```


TODO: describe

```
2429 \def\eql@stack@save@box{%
2430   \let\eql@stack@cell@container\eql@cell@container
2431   \edef\eql@stack@restore{%
2432     \eql@stack@save\let\eql@stack@cell@container\eql@cell@container
2433     \eql@stack@save@reg\eql@row@
2434   }%
2435 }
```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```
2436 \def\eql@measure@init#1#2{%
2437   \eql@dimensions@reset
2438   \let\eql@display@container\@empty
2439   \eql@numbering@measure@init
2440   \eql@row@\z@
2441   \eql@totalheight@\z@
2442   \eql@totalrows@\@M
2443   \eql@line@prevdepth@-\@m\p@
2444   \eql@line@interline@\z@
2445   \tabskip\z@skip
2446   \everycr{\noalign{%
2447     \global\advance\eql@row@\@ne
2448     #1%
2449   }}%
2450   \eql@punct@top@set
2451   \global\let\eql@interline@container\eql@interline@container@clear
2452   \eql@measure@savestate
2453   \eql@multi@cr@let{#2}%
2454 }
```

TODO: describe

```
2455 \def\eql@measure@tag{%
2456   \eql@tagwidth@\z@
2457   \ifdefined\eql@numbering@multi
2458     \if@eqnsw
2459       \eql@tags@container
2460       \eql@tagbox@make\eql@composetag@measure
2461       \ifdefined\eql@tagpos@reserve\else
2462         \eql@tagwidth@\z@
2463       \fi
2464     \fi
2465   \fi
2466 }
```

TODO: describe

```
2467 \def\eql@measure@endrow{%
2468   \ifdim\eql@line@prevdepth@=-\@m\p@\else
```

```

2469 \dimen@ \dimexpr \baselineskip - \eql@line@height@ - \eql@line@prevdepth@ \relax
2470 \ifdim \dimen@ < \lineskiplimit
2471 \dimen@ \lineskip
2472 \fi
2473 \advance \eql@line@interline@ \dimen@
2474 \fi
2475 \eql@dimensions@endrow
2476 \ifdim \eql@tagwidth@ > \eql@tagwidth@max@
2477 \global \eql@tagwidth@max@ \eql@tagwidth@
2478 \fi
2479 \ifdim \eql@tagwidth@ > \z@
2480 \global \advance \eql@tagrows@ \@ne
2481 \fi
2482 \global \advance \eql@totalheight@ \dimexpr
2483 \eql@line@interline@ + \eql@line@height@ + \eql@line@depth@
2484 \global \eql@line@interline@ \z@
2485 \global \eql@line@prevdepth@ \eql@line@depth@
2486 }

```

TODO: describe

```

2487 \def \eql@measure@close{%
2488 \advance \eql@row@ - \tw@
2489 \eql@totalrows@ \eql@row@
2490 \ifnum \eql@totalrows@ > \z@
2491 \eql@dimensions@get \@ne
2492 \eql@topheight@ \dimexpr \eql@line@height@ + \eql@line@interline@ \relax
2493 \eql@dimensions@get \eql@totalrows@
2494 \eql@bottomdepth@ \eql@line@depth@
2495 \fi
2496 \eql@numbering@measure@blocktag
2497 \begingroup
2498 \eql@tags@container
2499 \if@eqnsw
2500 \eql@tagbox@make \eql@composetag@measure
2501 \ifdefined \eql@tagpos@reserve \else
2502 \eql@tagwidth@ \z@
2503 \fi
2504 \eql@dimensions@saveblocktag
2505 \else
2506 \eql@dimensions@savenoblocktag
2507 \eql@numbering@warnunused
2508 \fi
2509 \endgroup
2510 \eql@dimensions@get \z@
2511 \eql@measure@restorestate
2512 }

```

measure@restorestate

eql@measure@savestate

```

2513 \let \eql@measure@restorestate \empty
2514 \def \eql@measure@savestate{%
2515 \begingroup
2516 \def \@elt##1{%
2517 \global \csname c@##1\endcsname \the \csname c@##1\endcsname}%
2518 \global \edef \@gtempa{\cl@@ckpt}%
2519 \endgroup
2520 \let \eql@measure@restorestate \@gtempa
2521 }

```

8.2 Line Breaks

TODO: describe

`\eqL@multi@cr`

```
2522 \def\eqL@multi@cr{%
2523   \let\eqL@punct@term\eqL@false
2524   \let\eqL@class@rel@composed\@empty
2525   \eqL@ampprotect\eqL@multi@cr@test\eqL@multi@cr@process}
```

TODO: describe

```
2526 \def\eqL@multi@cr@test@setopt{%
2527   \let\eqL@multi@cr@test\eqL@multi@cr@testopt}
2528 \def\eqL@multi@cr@test@setall{%
2529   \let\eqL@multi@cr@test\eqL@multi@cr@testall}
```

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

```
2530 \def\eqL@multi@cr@testopt#1{\eqL@teststaropt@tight
2531   {\eqL@displaybreak@star\@M\eqL@multi@cr@testopt@set{#1}}
2532   {\eqL@multi@cr@testopt@set{#1}}{Opt}}
2533 \def\eqL@multi@cr@testopt@set#1[#2]{\eqL@vspace@add{#2}#1}
```

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

`\lti@cr@testall@parse`

```
2534 \def\eqL@multi@cr@testall{\eqL@parseopt@cr\eqL@multi@cr@testall@parse}
2535 \def\eqL@multi@cr@testall@parse{%
2536   \ifx\eqL@parseopt@token%
2537     \let\eqL@parseopt@next\eqL@multi@cr@parse@vspace
2538   \fi
2539   \ifx\eqL@parseopt@token*%
2540     \let\eqL@parseopt@next\eqL@multi@cr@parse@star
2541   \fi
2542   \ifx\eqL@parseopt@token.%
2543     \let\eqL@parseopt@next\eqL@parseopt@punctpass
2544   \fi
2545   \ifx\eqL@parseopt@token,%
2546     \let\eqL@parseopt@next\eqL@parseopt@punctpass
2547   \fi
2548   \ifx\eqL@parseopt@token~%
2549     \let\eqL@parseopt@next\eqL@parseopt@punctpass
2550   \fi
2551   \ifx\eqL@parseopt@token'%
2552     \let\eqL@parseopt@next\eqL@parseopt@punctnext
2553   \fi
2554   \ifx\eqL@parseopt@token!%
2555     \let\eqL@parseopt@next\eqL@parseopt@punctterm
2556   \fi
2557   \ifx\eqL@parseopt@token/%
2558     \let\eqL@parseopt@next\eqL@multi@cr@parse@break
2559   \fi
2560   \ifx\eqL@parseopt@token=%
2561     \let\eqL@parseopt@next\eqL@parseopt@relsyb
2562   \fi
2563   \ifx\eqL@parseopt@token;%
2564     \let\eqL@parseopt@next\eqL@parseopt@relcont
2565   \fi
```

```

2566 \ifx\eql@parseopt@token:%
2567 \let\eql@parseopt@next\eql@parseopt@relstart
2568 \fi
2569 \ifx\eql@parseopt@token|%
2570 \let\eql@parseopt@next\eql@multi@cr@parse@rel
2571 \fi
2572 \ifx\eql@parseopt@token&%
2573 \let\eql@parseopt@next\eql@parseopt@end
2574 \fi
2575 }
2576 \def\eql@multi@cr@parse@vspace[#1]{\eql@vspace@add{#1}\eql@parseopt@peek}
2577 \def\eql@multi@cr@parse@star#1{\eql@displaybreak@star\@M\eql@parseopt@peek}
2578 \def\eql@multi@cr@parse@break{\numbernext\eql@parseopt@punctclear}
2579 \def\eql@multi@cr@parse@rel#1#2{%
2580 \def\eql@tmp{#2}%
2581 \ifx\eql@tmp\eql@relax\else
2582 \eql@punct@next@clear
2583 \ifdefined\eql@multi@cr@relnext\numbernext\fi
2584 \fi
2585 \ifdefined\eql@multi@linesmode
2586 \ifx\eql@tmp\@empty
2587 \def\eql@class@rel@composed{\eql@shape@cont}%
2588 \else
2589 \def\eql@class@rel@composed{\eql@shape@rel#2}%
2590 \fi
2591 \else
2592 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
2593 \fi
2594 \eql@parseopt@end}

```

eql@multi@cr@process

```

2595 \def\eql@multi@cr@process{%
2596 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
2597 \edef\eql@tmp{%
2598 \unexpanded{%
2599 \eql@multi@endline
2600 \cr
2601 \eql@multi@cr@interline
2602 }%
2603 \unexpanded\expandafter{\eql@class@rel@composed}%
2604 }%
2605 \eql@tmp
2606 }

```

```

2607 \def\eql@multi@cr@interline{%
2608 \noalign{%
2609 \eql@interline@container
2610 \eql@display@injectbefore
2611 \ifnum\eql@displaybreak@pen@=\@MM
2612 \penalty\interdisplaylinepenalty
2613 \else
2614 \penalty\eql@displaybreak@pen@
2615 \fi
2616 \vskip\eql@vspaceskip@
2617 \global\advance\eql@line@interline@\eql@vspaceskip@
2618 \eql@display@injectafter
2619 \global\let\eql@interline@container\eql@interline@container@clear
2620 }%

```

2621 }

`\eql@multi@cr@let`

```
2622 \def\eql@multi@cr@let#1{%
2623   \let\\\eql@multi@cr
2624   \let\eql@multi@endline#1%
2625 }
```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```
2626 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2627 \eql@amsmath@let\intertext\eql@intertext@default
```

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

```
2628 \def\eql@intertext@process{%
2629   \eql@multi@endline
2630   \cr
2631   \ifmeasuring@
2632     \expandafter\@gobble
2633   \else
2634     \expandafter\eql@intertext@print
2635   \fi
2636 }
```

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

```
2637 \def\eql@intertext@print#1{%
2638   \noalign{%
2639     \eql@display@halign@end
2640     \let\eql@skip@force@below\z@
2641     \let\eql@skip@force@above\z@
2642     \eql@setkeys{intertext}\eql@intertext@opt
2643     \openup-\eql@spread@
2644     \penalty\postdisplaypenalty
2645     \ifcase\eql@skip@force@below\relax
2646       \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2647     \or
2648       \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2649     \or
2650       \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2651     \or
2652       \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2653     \or
2654       \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2655     \or
2656       \advance\eql@vspaceskip@\z@skip
2657     \or
2658       \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2659     \or
2660       \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2661     \fi
2662     \vskip\eql@vspaceskip@
2663     \global\let\eql@interline@container\eql@interline@container@clear
```

```

2664 \vbox{%
2665 \parboxrestore
2666 \ifdim
2667 \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2668 \else
2669 \parshape\@ne
2670 \@totalleftmargin\linewidth
2671 \fi
2672 \noindent
2673 \prevgraf\eq\@prevgraf@
2674 \ignorespaces
2675 #1%
2676 \par
2677 \global\eq\@prevgraf@\prevgraf
2678 }%
2679 \penalty\predisplaypenalty
2680 \ifcase\eq\@skip@force@above\relax
2681 \vskip\glueexpr\eq\@skip@long@above\relax
2682 \or
2683 \vskip\glueexpr\eq\@skip@short@above\relax
2684 \or
2685 \vskip\glueexpr\eq\@skip@cont@above\relax
2686 \or
2687 \vskip\glueexpr\eq\@skip@par@above\relax
2688 \or
2689 \vskip\glueexpr\eq\@skip@top@above\relax
2690 \or
2691 \vskip\z@skip
2692 \or
2693 \vskip\glueexpr\eq\@skip@med@above\relax
2694 \or
2695 \vskip\glueexpr\eq\@skip@custom@above\relax
2696 \fi
2697 % \eq\@prevdepth@\maxdimen
2698 \eq\@prevdepth@\z@
2699 \eq\display@halign@start
2700 }
2701 }

```

TODO: describe

```

2702 \newenvironment{eq\intertext}{%
2703 \eq\testopt@tight\eq\intertext@{}%
2704 }{%
2705 \aftergroup\eq\intertext@after
2706 \ignorespacesafterend
2707 }

```

TODO: describe

```

2708 \def\eq\intertext@env{intertext}
2709 \def\eq\intertext@[#1]{%
2710 \global\def\eq\intertext@opt{#1}%
2711 \ifx\@currenvir\eq\intertext@env
2712 \def\eq\scan@call{\eq\intertext@inject\eq\scan@end}%
2713 \expandafter\eq\scan@env
2714 \else
2715 \expandafter\eq\intertext@process
2716 \fi
2717 }

```

TODO: describe

```
2718 \def\eql@intertext@inject{%
2719   \global\edef\eql@intertext@after{%
2720     \noexpand\eql@intertext@process{%
2721       \ifx\eql@scan@body\eql@scan@body@dump
2722         \eql@scan@body@dump
2723       \else
2724         \noexpand\scantokens{\eql@scan@body@dump}%
2725       \fi
2726     }%
2727   }%
2728 }
```

8.4 Line Marks

TODO: describe

```
2729 \def\eql@markline@pos@below{below}
2730 \def\eql@markline@pos@bottom{bottom}
2731 \def\eql@markline@pos@baseline{baseline}
2732 \let\eql@markline@pos\eql@markline@pos@baseline
2733 \let\eql@markline@shift\z@
2734 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2735 \def\eql@markline@symbol{}
```

TODO: describe

```
2736 \def\eql@markline@select#1{%
2737   \let\eql@markline@shift\z@
2738   \eql@setkeys{markline}{#1}%
2739   \eql@markline@print
2740 }
```

TODO: describe

```
2741 \def\eql@markline@print{%
2742   \dimen@=\dimexpr\eql@markline@shift\relax
2743   \ifx\eql@markline@pos\eql@markline@pos@below
2744     \ifdim\dimen@=\z@\else
2745       \penalty\@M
2746       \vskip-\dimen@
2747     \fi
2748     \nointerlineskip
2749     \penalty\@M
2750     \vbox{\hfill\hbox{\eql@markline@symbol}}%
2751   \else
2752     \ifx\eql@markline@pos\eql@markline@pos@baseline
2753       \advance\dimen@\prevdepth
2754     \fi
2755     \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}%
2756     \dimen@\prevdepth
2757     \ht\z@\z@
2758     \dp\z@\z@
2759     \nointerlineskip
2760     \penalty\@M
2761     \vbox{\hfill\box\z@}%
2762     \prevdepth\dimen@
2763   \fi
}
```

2764 }

TODO: describe

```

2765 \def\eql@markline@inject#1{%
2766   \let\eql@markline@push\eql@false
2767   \ifx\eql@markline@pos\eql@markline@pos@below\else
2768     \ifdefined\eql@tagsleft\else
2769       \ifx\eql@equations@main\eql@multi@main
2770         \ifdefined\eql@numbering@multi
2771           \if@eqnsw
2772             \let\eql@markline@push\eql@true
2773           \fi
2774         \else
2775           \ifnum\eql@row@=\eql@tagpos@row@
2776             \let\eql@markline@push\eql@true
2777           \fi
2778         \fi
2779       \else
2780         \if@eqnsw
2781           \let\eql@markline@push\eql@true
2782         \fi
2783       \fi
2784     \fi
2785   \fi
2786   \ifdefined\eql@markline@push
2787     \global\eql@append\eql@interline@container{%
2788       \eql@append\eql@display@injectbefore{\eql@markline@select{push,#1}}}%
2789   \else
2790     \global\eql@append\eql@interline@container{%
2791       \eql@append\eql@display@injectbefore{\eql@markline@select{#1}}}%
2792   \fi
2793 }
```

TODO: describe

```

2794 \def\eql@markline@amsthm@opt[#1]{\eql@markline@inject{qed,#1}}
2795 \def\eql@markline@amsthm@staropt[#1]{\eql@markline@inject{qed,push,#1}}
2796 \def\eql@markline@amsthm@qed{\eql@teststaropt@tight
2797   \eql@markline@amsthm@staropt\eql@markline@amsthm@opt{}}
2798 \def\eql@markline@amsthm@register#1{\eql@letcs{#1@qed}\eql@markline@amsthm@qed}
2799 \def\eql@markline@amsthm@move#1#2{%
2800   \AddToHook{package/amsthm/after}{%
2801     \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:

```

2802 \newcount\eql@shape@pos@
2803 \newdimen\eql@shape@amount@
```



```
2804 \let\eq@shape@lastrow\eq@false
```

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

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

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

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

```
2805 \newdimen\eq@marginleft@
2806 \newdimen\eq@marginright@
2807 \newdimen\eq@marginleft@min@
2808 \newdimen\eq@centeroffset@
```

9.2 Shape Schemes

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

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

```
2809 \def\eq@shape@tab@default{default}
2810 \def\eq@shape@tab@left{left}
2811 \def\eq@shape@tab@center{center}
2812 \def\eq@shape@tab@right{right}
2813 \def\eq@shape@tab@first{first}
2814 \def\eq@shape@tab@hanging{hanging}
2815 \def\eq@shape@tab@steps{steps}
```

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

```
2816 \let\eq@shape@tab@def\eq@shape@tab@default
2817 \let\eq@shape@tab@\eq@shape@tab@default
2818 \let\eq@shape@tab@l\eq@shape@tab@left
2819 \let\eq@shape@tab@c\eq@shape@tab@center
2820 \let\eq@shape@tab@r\eq@shape@tab@right
2821 \let\eq@shape@tab@rc\eq@shape@tab@first
2822 \let\eq@shape@tab@indent\eq@shape@tab@first
2823 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2824 \let\eq@shape@tab@lc\eq@shape@tab@hanging
2825 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2826 \let\eq@shape@tab@lcr\eq@shape@tab@steps
```

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

```
2827 \let\eq@shape@mode\eq@shape@tab@default
```

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

```
2828 \def\eq@shape@set#1{%
2829   \ifcsname eq@shape@tab@#1\endcsname
2830     \expandafter\let\expandafter\eq@shape@mode
2831       \csname eq@shape@tab@#1\endcsname
2832   \else
2833     \eq@error{shape '#1' unknown: setting to default}%
2834     \let\eq@shape@mode\eq@shape@tab@default
2835   \fi
2836 }
```

`\eq@shape@layoutcenter@...` Define the uniform shape schemes `left`, `center`, `right` and `default` for the central and

`\eq@shape@layoutleft@...` left alignment layout. The scheme functions determine the desired alignment and indentation for the current row:

```

2837 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
2838 \def\eq@shape@layoutcenter@center{\eq@shape@pos@one\eq@shape@amount@z@}
2839 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2840 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2841 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
2842 \def\eq@shape@layoutleft@center{\eq@shape@pos@one\eq@shape@amount@z@}
2843 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2844 \let\eq@shape@layoutleft@default\eq@shape@layoutleft@left

```

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

```

2845 \def\eq@shape@layoutcenter@first{%
2846   \eq@shape@pos@z@
2847   \eq@shape@amount@z@
2848   \ifnum\eq@totalrows@>\@ne
2849     \ifnum\eq@row@=\@ne
2850       \eq@shape@amount@\eq@indent@
2851     \fi
2852   \fi
2853 }
2854 \def\eq@shape@layoutleft@first{%
2855   \eq@shape@pos@z@
2856   \eq@shape@amount@z@
2857   \ifnum\eq@totalrows@>\@ne
2858     \ifnum\eq@row@=\@ne
2859       \eq@shape@amount@\eq@indent@
2860     \fi
2861   \fi
2862 }

```

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:

```

2863 \def\eq@shape@layoutcenter@hanging{%
2864   \eq@shape@pos@z@
2865   \eq@shape@amount@\eq@indent@
2866   \ifnum\eq@totalrows@>\@ne
2867     \ifnum\eq@row@=\@ne
2868       \eq@shape@amount@z@
2869     \fi
2870   \fi
2871 }
2872 \def\eq@shape@layoutleft@hanging{%
2873   \eq@shape@pos@z@
2874   \eq@shape@amount@z@
2875   \ifnum\eq@totalrows@>\@ne
2876     \ifnum\eq@row@=\@ne
2877       \eq@shape@amount@-\eq@indent@
2878     \fi
2879   \fi
2880 }

```

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:

```

2881 \def\eq@shape@layoutcenter@steps{%
2882   \eq@shape@amount@z@

```

```

2883 \eq@shape@pos@\@ne
2884 \ifnum\eq@totalrows@>\@ne
2885   \ifnum\eq@row@=\@ne
2886     \eq@shape@pos@\z@
2887   \fi
2888   \ifnum\eq@row@=\eq@totalrows@
2889     \eq@shape@pos@\tw@
2890   \fi
2891 \fi
2892 }
2893 \def\eq@shape@layoutleft@steps{%
2894   \eq@shape@pos@\z@
2895   \eq@shape@amount@\z@
2896   \ifnum\eq@totalrows@>\@ne
2897     \ifnum\eq@row@=\@ne
2898       \eq@shape@amount@-\eq@indent@
2899     \fi
2900     \ifnum\eq@row@=\eq@totalrows@
2901       \eq@shape@amount@\eq@indent@
2902     \fi
2903 \fi
2904 }

```

`\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`:

```

2905 \let\eq@shape@eval\@undefined
2906 \def\eq@shape@select{%
2907   \expandafter\let\expandafter\eq@shape@eval
2908   \csname eq@shape%
2909     @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2910     @\eq@shape@mode\endcsname
2911 }

```

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

```

2912 \protected\def\eq@shape@alignleft{%
2913   \global\eq@append\eq@cell@container{\eq@shape@pos@\z@}%
2914   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2915 \protected\def\eq@shape@aligncenter{%
2916   \global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}%
2917   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2918 \protected\def\eq@shape@alignright{%
2919   \global\eq@append\eq@cell@container{\eq@shape@pos@\tw@}%
2920   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2921 \def\eq@shape@align@testpar#1{%
2922   \eq@ifstar@tight{#1[\eq@indent@]}%
2923   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2924   {\eq@testopt@tight{#1}\z@}}
2925 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2926 \protected\def\eq@shape@alignamount{%
2927   \eq@ampprotecttwo\eq@ifstar@tight
2928   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2929 \def\eq@shape@alignamount@add#1{%
2930   \global\eq@appendexpand\eq@cell@container{

```

```

2931 \advance\eq\shape@amount@\the\glueexpr#1\relax\relax}}
2932 \def\eq\shape@alignamount@set#1{%
2933 \global\eq\appendexpand\eq\cell@container{%
2934 \eq\shape@amount@\the\glueexpr#1\relax\relax}}
2935 \def\eq\shape@align@enable{%
2936 \let\shoveleft\eq\shape@alignleft
2937 \let\shovecenter\eq\shape@aligncenter
2938 \let\shoveright\eq\shape@alignright
2939 \let\shoveby\eq\shape@alignamount
2940 }

```

TODO: describe

```

2941 \protected\def\eq\shape@align@default{%
2942 \eq\warn@here{\shove...}%
2943 \eq\ampprotect\eq\shape@align@testpar\eq@gobbleopt}
2944 \protected\def\eq\shape@alignamount@default{%
2945 \eq\warn@here{\shove...}%
2946 \eq\ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2947 \def\eq\shape@align@disable{%
2948 \let\shoveleft\eq\shape@align@default
2949 \let\shovecenter\eq\shape@align@default
2950 \let\shoveright\eq\shape@align@default
2951 \let\shoveby\eq\shape@alignamount@default
2952 }

```

9.3 Width Data

width@block@ (*dimen*)

```

2953 \newdimen\eq\tagwidth@block@
2954 \newdimen\eq\tagheight@block@
2955 \newdimen\eq\tagdepth@block@

```

$\text{\eq@dimensions@tab}$ **TODO:** new

```

2956 \let\eq@dimensions@tab\@empty

```

$\text{\eq@dimensions@reset}$

```

2957 \def\eq@dimensions@reset{%
2958 \let\eq@dimensions@tab\@empty
2959 \eq\tagwidth@max@\z@
2960 \eq\tagrows@\z@
2961 }

```

$\text{\eq@dimensions@add}$

```

2962 \def\eq@dimensions@add#1{%
2963 \global\eq\appendexpand\eq@dimensions@tab{#1}%
2964 }

```

$\text{\eq@dimensions@addreg}$

```

2965 \def\eq@dimensions@addreg#1{#1\the#1\relax}

```

$\text{\eq@dimensions@startrow}$

```

2966 \def\eq@dimensions@startrow{%

```

```

2967 \eqldimensions@add{\eqldimensions@addreg\eqldrow}%
2968 }

@dimensions@savecell

2969 \def\eqldimensions@savecell{%
2970 \eqldimensions@add{%
2971 \eqldimensions@addreg\eqldshape@pos@
2972 \eqldimensions@addreg\eqldcellwidth@
2973 \eqldimensions@addreg\eqldshape@amount@
2974 \noexpand\eqldimensions@cellcall
2975 }%
2976 }

l@dimensions@savesep

2977 \def\eqldimensions@savesep{%
2978 \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2979 }

ql@dimensions@endrow

2980 \def\eqldimensions@endrow{%
2981 \eqldimensions@add{,%
2982 \eqldimensions@addreg\eqldtagwidth@
2983 \eqldimensions@addreg\eqldline@height@
2984 \eqldimensions@addreg\eqldline@depth@
2985 \eqldimensions@addreg\eqldline@interline@
2986 ;}%
2987 }

ensions@saveblocktag

2988 \def\eqldimensions@saveblocktag{%
2989 \eqldimensions@add{\eqldrow@0\relax,%
2990 \eqldtagwidth@block@\the\eqldtagwidth@\relax
2991 \eqldtagheight@block@\the\ht\eqldtagbox@\relax
2992 \eqldtagdepth@block@\the\dp\eqldtagbox@\relax
2993 \eqldimensions@addreg\eqldtagpos@shift@
2994 \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
2995 \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
2996 ;}%
2997 \global\eqldtagwidth@max@\eqldtagwidth@
2998 \global\eqldtaggrows@\@ne
2999 }

sions@savenoblocktag

3000 \def\eqldimensions@savenoblocktag{%
3001 \eqldimensions@add{\eqldrow@0\relax,;%
3002 }

\eqldimensions@for

3003 \def\eqldimensions@for#1{%
3004 \def\eqldimensions@forcall{#1}%
3005 \expandafter\eqldimensions@forstep\eqldimensions@tab
3006 }

```

l@dimensions@forstep

```

3007 \def\eql@dimensions@forstep\eql@row@#1\relax#2,##3;{%
3008   \eql@row@#1\relax
3009   \ifnum\eql@row@=\z@\else
3010     #3%
3011     \def\eql@dimensions@cells{##2}%
3012     \eql@dimensions@forall
3013     \expandafter\eql@dimensions@forstep
3014   \fi
3015 }

```

\eql@dimensions@get

```

3016 \def\eql@dimensions@get#1{%
3017   \eql@row@#1\relax
3018   \expandafter\eql@dimensions@getdef\expandafter{\the\eql@row@}%
3019   \expandafter\eql@dimensions@getparse\eql@dimensions@tab\@nil
3020 }

```

ql@dimensions@getdef

```

3021 \def\eql@dimensions@getdef#1{%
3022   \def\eql@dimensions@getparse
3023     ##1\eql@row@#1\relax##2,##3;##4\@nil{%
3024     ##3%
3025     \def\eql@dimensions@cells{##2}%
3026   }%
3027 }

```

\eql@colwidth@tab

```

3028 \let\eql@colwidth@tab\@empty

```

\eql@colwidth@get

```

3029 \def\eql@colwidth@get#1{%
3030   \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
3031 }

```

\eql@colwidth@save

```

3032 \def\eql@colwidth@save#1{%
3033   \edef\eql@colwidth@tab{%
3034     \noexpand\or\the#1%
3035     \unexpanded\expandafter{\eql@colwidth@tab}%
3036   }%
3037 }

```

\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@:

```

3038 \def\eql@dimensions@calc{%
3039   \eql@column@\z@
3040   \eql@line@pos@\z@

```

```

3041 \eq\line@possep@\z@
3042 \eq\line@avail@\eq\totalwidth@
3043 \eq\line@availsep@\eq\intercolumnsep@
3044 \eq\line@width@\z@
3045 \eq\line@widthsep@\z@
3046 \let\eq\dimensions@cellcall\eq\dimensions@calc@call
3047 \let\eq\dimensions@sepcall\eq\dimensions@calc@callsep
3048 \eq\dimensions@cells
3049 }

```

`\eq\dimensions@calc@callsep` Callback for each intercolumn space.

```

3050 \def\eq\dimensions@calc@callsep{%
3051   \advance\eq\line@possep@\@ne
3052 }%

```

`\eq\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?!)

```

3053 \def\eq\dimensions@calc@call{%
3054   \advance\eq\column@\@ne
3055   \ifnum\eq\totalcolumns@=\@ne
3056     \dimen@\eq\totalwidth@
3057   \else
3058     \dimen@\eq\colwidth@get\eq\column@\relax
3059   \fi
3060   \ifdim\eq\cellwidth@>\z@
3061     \ifdim\eq\line@width@=\z@
3062       \eq\line@avail@\eq\line@pos@
3063       \eq\line@availsep@\eq\line@possep@
3064       \ifcase\eq\shape@pos@
3065       \or
3066         \advance\eq\line@avail@\dimexpr
3067           (\dimen@-\eq\cellwidth@+\eq\centeroffset@)/\tw@\relax
3068       \or
3069         \advance\eq\line@avail@\dimexpr\dimen@-\eq\cellwidth@\relax
3070       \fi
3071       \advance\eq\line@avail@\eq\shape@amount@
3072     \fi
3073     \eq\line@width@\eq\line@pos@
3074     \eq\line@widthsep@\eq\line@possep@
3075     \ifcase\eq\shape@pos@
3076     \advance\eq\line@width@\eq\cellwidth@
3077     \or
3078     \advance\eq\line@width@\dimexpr
3079       (\dimen@+\eq\cellwidth@+\eq\centeroffset@)/\tw@\relax
3080     \or
3081     \advance\eq\line@width@\dimen@
3082     \fi
3083     \advance\eq\line@width@\eq\shape@amount@
3084   \fi
3085   \advance\eq\line@pos@\dimen@
3086 }

```

9.4 Best Line Selection

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

```
3087 \let\eql@numbering@best@auto\eql@false
```

`@g@best@row@` (*counter*)

`@g@best@space@` (*dimen*)

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

```
3088 \newcount\eql@numbering@best@row@
```

```
3089 \newdimen\eql@numbering@best@space@
```

```
3090 \let\eql@numbering@best@use\eql@false
```

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

```
3091 \def\eql@numbering@best@find{%
3092   \eql@numbering@best@row@ \z@
3093   \eql@numbering@best@space@ \z@
3094   \eql@dimensions@for{%
3095     \eql@dimensions@calc
3096     \ifdefined\eql@tagsleft
3097       \dimen@ \eql@line@avail@
3098     \else
3099       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
3100     \fi
3101     \ifdim \dimen@ > \eql@numbering@best@space@
3102       \eql@numbering@best@row@ \eql@row@
3103       \eql@numbering@best@space@ \dimen@
3104     \fi
3105   }%
3106   \ifnum \eql@numbering@best@row@ > \z@
3107     \eql@tagpos@row@ \eql@numbering@best@row@
3108     \let\eql@tagpos@continuous\eql@false
3109     \eql@tagpos@prevrow@ \z@
3110   \fi
3111 }
```

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

```
3112 \def\eql@numbering@best@test#1{%
3113   \eql@dimensions@get#1%
3114   \eql@dimensions@calc
3115   \ifdefined\eql@tagsleft
3116     \dimen@ \dimexpr \eql@line@avail@
3117       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
3118   \else
3119     \dimen@ \dimexpr \displaywidth@ - \eql@line@width@
3120       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
3121   \fi
3122   \ifdim \dimen@ < \eql@tagwidth@block@
3123     \let\eql@numbering@best@use\eql@true
3124   \fi
3125 }
```

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

```
3126 \def\eql@numbering@best@eval{%
3127   \ifdefined\eql@numbering@best@auto
3128     \ifdefined\eql@numbering@best@use\else
3129     \ifdefined\eql@numbering@multi\else
```



```

3130         \ifnum\eql@tagpos@row@>\z@
3131             \eql@numbering@best@test\eql@tagpos@row@
3132         \fi
3133         \ifnum\eql@tagpos@prevrow@>\z@
3134             \eql@numbering@best@test\eql@tagpos@prevrow@
3135         \fi
3136     \fi
3137 \fi
3138 \fi
3139 \ifdefined\eql@numbering@best@use
3140     \eql@numbering@best@find
3141 \fi
3142 }

```

9.5 Tag Margin

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

djust@calc@tagmargin

```

3143 \def\eql@adjust@calc@tagmargin{%
3144     \ifdefined\eql@tagmargin@val
3145         \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
3146     \else
3147         \eql@tagmargin@\eql@tagwidth@max@
3148         \ifdim\eql@tagmargin@>\z@
3149             \advance\eql@tagmargin@-\eql@tagsepmin@
3150         \fi
3151     \fi

3152     \dimen@\eql@tagrows@\p@
3153     \ifnum\eql@totalrows@=\@ne
3154         \ifnum\eql@tagrows@=\@ne
3155             \advance\dimen@1sp\relax
3156         \fi
3157     \fi
3158     \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
3159         \eql@tagmargin@\z@
3160     \fi

3161     \@tempdima\dimexpr\displaywidth
3162         -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
3163     \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
3164     \ifdim\@tempdimb>\z@
3165         \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
3166             \eql@tagmargin@\z@
3167         \fi
3168     \fi
3169 }

```

9.6 Single Column

ql@adjust@calc@lines

```

3170 \def\eql@adjust@calc@lines{%
3171     \eql@totalcolumns@\@ne

```

```

3172 \eq@intercolumns@z@
3173 \eq@colsep@z@
3174 \ifdefined\eq@layoutleft
3175 \eq@marginleft@glueexpr\eq@layoutleftmargin\relax
3176 \eq@marginleft@min@glueexpr\eq@layoutleftmarginmin\relax
3177 \ifdim\eq@marginleft@<\eq@marginleft@min@
3178 \eq@marginleft@\eq@marginleft@min@
3179 \fi
3180 \dimen@glueexpr\eq@layoutleftmarginmax\relax
3181 \ifdim\eq@marginleft@>\dimen@
3182 \eq@marginleft@\dimen@
3183 \fi
3184 \eq@marginright@z@
3185 \eq@centeroffset@z@
3186 \else
3187 \eq@adjust@calc@tagmargin
3188 \ifdefined\eq@paddingleft@val
3189 \eq@marginleft@\dimexpr
3190 (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3191 -\glueexpr\eq@paddingleft@val\relax\relax
3192 \ifdim\eq@marginleft@<z@
3193 \eq@marginleft@z@
3194 \fi
3195 \else
3196 \eq@marginleft@z@
3197 \fi
3198 \ifdefined\eq@paddingright@val
3199 \eq@marginright@\dimexpr
3200 (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3201 -\glueexpr\eq@paddingright@val\relax\relax
3202 \ifdim\eq@marginright@<z@
3203 \eq@marginright@z@
3204 \fi
3205 \else
3206 \eq@marginright@z@
3207 \fi
3208 \ifdim\eq@tagmargin@>z@
3209 \ifdefined\eq@tagsleft
3210 \ifdim\eq@marginleft@<\eq@tagsepmin@
3211 \eq@marginleft@\eq@tagsepmin@
3212 \fi
3213 \advance\eq@marginleft@\eq@tagmargin@
3214 \advance\eq@centeroffset@\eq@tagmargin@
3215 \else
3216 \ifdim\eq@marginright@<\eq@tagsepmin@
3217 \eq@marginright@\eq@tagsepmin@
3218 \fi
3219 \advance\eq@marginright@\eq@tagmargin@
3220 \advance\eq@centeroffset@-\eq@tagmargin@
3221 \fi
3222 \fi
3223 \eq@marginleft@min@z@
3224 \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@
3225 \ifdefined\eq@tagsleft+ \else -\fi \eq@tagmargin@ \relax
3226 \fi

3227 \eq@totalwidth@\dimexpr\displaywidth
3228 -\eq@marginleft@-\eq@marginright@ \relax
3229 }

```

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:

```
3230 \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:

```
3231 \ifnum\eql@totalcolumns@<\thr@@
3232   \eql@totalcolumns@=\tw@
3233   \let\eql@columns@fulllength\eql@false
3234 \fi
```

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

```
3235 \eql@intercolumns@=\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
3236 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

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

```
3237 \ifdefined\eql@layoutleft
3238   \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
3239   \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
3240   \ifdim\eql@marginleft@<\eql@marginleft@min@
3241     \eql@marginleft@\eql@marginleft@min@
3242   \fi
3243 \else
```

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

```
3244   \eql@adjust@calc@tagmargin
3245   \ifdefined\eql@columns@fulllength
3246     \ifdim\eql@tagmargin@>\z@
3247       \advance\eql@tagmargin@\eql@tagsepmin@
3248     \fi
3249   \fi
3250   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
3251     -\eql@intercolumns@\eql@colsepmin@\relax
3252     \eql@tagmargin@\z@
3253   \fi
3254   \eql@marginleft@min@\z@
3255 \fi
```

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

```
3256 \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:

```

3257 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
3258 \ifdefined\eql@layoutleft
3259 \advance\eql@colsep@-\eql@marginleft@
3260 \else
3261 \advance\eql@colsep@-\eql@tagmargin@
3262 \fi
3263 \count@\eql@intercolumns@
3264 \ifdefined\eql@columns@fulllength\else
3265 \ifdefined\eql@layoutleft
3266 \advance\count@\@ne
3267 \else
3268 \advance\count@\tw@
3269 \fi
3270 \fi
3271 \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:

```

3272 \ifdim\eql@colsep@<\eql@colsepmin@
3273 \eql@colsep@\eql@colsepmin@
3274 \else
3275 \ifdefined\eql@columns@fulllength\else
3276 \dimen@\glueexpr\eql@colsepmax@val\relax
3277 \ifdim\eql@colsep@>\dimen@
3278 \eql@colsep@\dimen@
3279 \fi
3280 \fi
3281 \fi
3282 \else

```

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

```

3283 \eql@colsep@\eql@colsepmin@
3284 \fi

```

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

```

3285 \ifdefined\eql@layoutleft

```

Set the default value:

```

3286 \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:

```

3287 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
3288 -\eql@intercolumns@\eql@colsep@\relax
3289 \ifdim\dimen@<\eql@marginleft@
3290 \ifdim\dimen@<\eql@marginleft@min@
3291 \eql@marginleft@\eql@marginleft@min@
3292 \else
3293 \eql@marginleft@\dimen@
3294 \fi
3295 \fi
3296 \fi
3297 \else

```

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

```

3298 \ifdefined\eql@columns@fulllength
3299 \eql@marginleft@z@

```

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

```

3300 \else
3301 \eql@marginleft@dimexpr(\displaywidth-\eql@totalwidth@
3302 -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@relax
3303 \ifdim\eql@marginleft@<z@
3304 \eql@marginleft@z@
3305 \fi
3306 \fi

```

Add tag margin in case of left tags:

```

3307 \ifdefined\eql@tagsleft
3308 \advance\eql@marginleft@\eql@tagmargin@
3309 \fi
3310 \fi

```

Find the best row for tag placement:

```

3311 \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:

```

3312 \ifdefined\eql@tagsleft
3313 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
3314 \else
3315 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
3316 \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

```

3317 \ifdefined\eql@numbering@multi
3318 \eql@dimensions@for{%
3319 \ifdim\eql@tagwidth@>z@
3320 \eql@dimensions@calc
3321 \eql@adjust@columns@test
3322 \fi
3323 }%
3324 \else
3325 \ifnum\eql@tagpos@row@>z@
3326 \ifnum\eql@tagpos@row@>\eql@totalrows@\else
3327 \eql@dimensions@get\eql@tagpos@row@
3328 \eql@tagwidth@\eql@tagwidth@block@
3329 \eql@dimensions@calc
3330 \eql@adjust@columns@test
3331 \fi
3332 \fi
3333 \ifnum\eql@tagpos@prevrow@>z@
3334 \eql@dimensions@get\eql@tagpos@prevrow@
3335 \eql@tagwidth@\eql@tagwidth@block@
3336 \eql@dimensions@calc
3337 \eql@adjust@columns@test
3338 \fi

```

```
3339 \fi
```

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

```
3340 \advance\eq@totalwidth@\dimexpr
3341 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
3342 }
```

Placement for Right Tags.

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

```
3343 \def\eq@adjust@columns@test@tagsright{%
```

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

```
3344 \@tempdima\dimexpr\displaywidth-\eq@linewidth-\eq@tagwidth@\relax
```

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

```
3345 \ifdim\@tempdima<\dimexpr
3346 \eq@marginleft@+\eq@linewidthsep@\eq@colsep@\relax
```

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

```
3347 \ifdim\@tempdima<\dimexpr
3348 \eq@marginleft@min@+\eq@linewidthsep@\eq@colsepmin@\relax\else
```

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

```
3349 \eq@adjust@columns@modify@tagsright
3350 \fi
3351 \fi
3352 }
```

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

```
3353 \def\eq@adjust@columns@modify@tagsright{%
```

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

```
3354 \ifnum\eq@linewidthsep@>\z@
3355 \dimen@\@tempdima
3356 \count@\eq@linewidthsep@
3357 \ifdefined\eq@layoutleft
3358 \advance\dimen@-\eq@marginleft@
3359 \else
3360 \ifdefined\eq@columns@fulllength\else
3361 \advance\count@\@ne
3362 \fi
3363 \fi
3364 \divide\dimen@\count@
```

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

```

3365 \ifdim\dimen@<\eq\colsep@
3366 \ifdim\dimen@<\eq\colsepmin@
3367 \eq\colsep@\eq\colsepmin@
3368 \else
3369 \eq\colsep@\dimen@
3370 \fi
3371 \fi
3372 \fi

```

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

```

3373 \dimen@\dimexpr\@tempdima-\eq\line@widthsep@\eq\colsep@\relax
3374 \ifdim\eq\marginleft@>\dimen@
3375 \eq\marginleft@\dimen@
3376 \fi
3377 }

```

Placement for Left Tags.

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

```

3378 \def\eq\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.

```

3379 \count@\numexpr\eq\intercolumns@-\eq\line@availsep@\relax
3380 \@tempdima\dimexpr\eq>tagwidth@-\eq\line@avail@\relax

```

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

```

3381 \ifdim\@tempdima>\dimexpr
3382 \eq\marginleft@+\eq\line@availsep@\eq\colsep@\relax

```

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

```

3383 \ifdim\eq>tagwidth@<%
3384 \ifdefined\eq\layoutleft
3385 \glueexpr\eq\layoutleftmarginmax\relax
3386 \else
3387 \displaywidth
3388 \fi

```

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

```

3389 \ifdim\@tempdima>\dimexpr
3390 \displaywidth-\eq\totalwidth@-\count@\eq\colsepmin@\relax\else

```

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

```

3391 \eq\adjust@columns@modify@tagsleft
3392 \fi
3393 \fi
3394 \fi
3395 }

```

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

```
3396 \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.

```
3397 \ifnum\count@>\z@
3398 \dimen@ \dimexpr\displaywidth-\eql@totalwidth@-\@tempdima\relax
3399 \ifdefined\eql@columns@fulllength\else
3400 \advance\count@\@ne
3401 \fi
3402 \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.

```
3403 \ifdim\dimen@<\eql@colsep@
3404 \ifdim\dimen@<\eql@colsepmin@
3405 \dimen@\eql@colsepmin@
3406 \fi
3407 \advance\dimen@-\eql@colsep@
3408 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3409 \advance\eql@colsep@\dimen@
3410 \fi
3411 \fi
```

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

```
3412 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3413 \ifdim\eql@marginleft@<\dimen@
3414 \eql@marginleft@\dimen@
3415 \fi
3416 }
```

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:

```
3417 \ifdefined\inf@bad\else%
3418 \newcount\inf@bad
3419 \inf@bad1000000\relax
3420 \fi
```

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

```
3421 \let\eql@restore@hfuzz\@empty
3422 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```


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

```
3423 \newcount\eqalignbadness@
3424 \newcount\eqtagbadness@
3425 \newcount\eqarrange@badness@
3426 \eqalignbadness@\inf@bad
3427 \eqtagbadness@\inf@bad
```

10.2 Arrangement Methods

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

```
3428 \def\eqarrange@try#1{%
3429   \ifdim#1>\dimexpr\displaywidth-\eqcellwidth@\relax
3430     \setbox\eqcellbox@\hbox to\displaywidth{%
3431       \unhbox\eqcellbox@\unkern\kern#1}%
3432     \eqarrange@badness@badness
3433   \else
3434     \eqarrange@badness@m@ne
3435   \fi
3436 }
```

`\eqarrange@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

```
3437 \def\eqarrange@print#1#2{%
3438   \eqrestore@hfuzz
3439   \if@eqnsw
3440     \ifdefined\eqtagsleft
3441       \eqtagbox@print@tagsleft
3442     \fi
3443   \fi
3444   \hbox to\displaywidth{%
3445     #1%
3446     \unhbox\eqcellbox@\unkern
3447     #2%
3448     \eqtagging@mathaddlast
3449   }%
3450   \if@eqnsw
3451     \ifdefined\eqtagsleft\else
3452       \eqtagbox@print@tagsright
3453     \fi
3454   \fi
3455 }
```

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

```
3456 \def\eqarrange@print@alignleft#1#2{%
3457   \eqdisplay@firstavail@set{\dimexpr#1\relax}%
3458   \eqarrange@print{\kern#1}{\kern#2}%
3459 }
```

```

3460 \def\eql@arrange@print@alignright#1#2{%
3461   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3462   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3463 }

3464 \def\eql@arrange@print@aligncenter#1{%
3465   \eql@display@firstavail@set{\dimexpr
3466     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3467   \ifdim#1>\z@
3468     \eql@arrange@print{\kern#1\hfil}{}%
3469   \else
3470     \eql@arrange@print{\hfil}{\kern-#1}%
3471   \fi
3472 }

```

`\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:

```

3473 \def\eql@arrange@init{%
3474   \eql@save@hfuzz
3475   \hfuzz\maxdimen
3476   \eql@shape@select
3477 }

```

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

```

3478 \def\eql@arrange@print@line{%
3479   \eql@tagging@tagaddbox
3480   \csname eql@arrange%
3481     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3482     @init\endcsname
3483   \csname eql@arrange%
3484     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3485     @\ifdefined\eql@tagpos@reserve
3486       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3487       notag\fi\endcsname
3488 }

```

10.3 Central Alignment

TODO: describe

```

3489 \def\eql@arrange@aligncenter@init{%
3490   \eql@tagging@aligncenter
3491   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3492     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3493 }

```

TODO: describe

```

3494 \def\eql@arrange@aligncenter@notag{%
3495   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3496     \ifdim\eql@line@offset@<\eql@marginleft@min@
3497       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3498     \else
3499       \eql@line@offset@
3500   \fi

```

```

3501 \eql@arrange@print@aligncenter\eql@line@offset@
3502 \else
3503 \ifdim\eql@line@offset@<\eql@marginleft@min@
3504 \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3505 \else
3506 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3507 \fi
3508 \fi
3509 }

```

TODO: describe

```

3510 \def\eql@arrange@aligncenter@tagsright{%
3511 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3512 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3513 \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3514 \else
3515 \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
3516 \fi
3517 \eql@arrange@print@aligncenter\eql@line@offset@
3518 \else
3519 \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
3520 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3521 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3522 \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3523 \else
3524 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3525 \fi
3526 \else
3527 \let\eql@tagpos@reserve\eql@false
3528 \eql@arrange@aligncenter@notag
3529 \fi
3530 \fi
3531 }

```

```

3532 \def\eql@arrange@aligncenter@tagsleft{%
3533 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3534 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3535 \ifdim\eql@line@offset@<\eql@tagwidth@
3536 \dimexpr\tw@\eql@tagwidth@-\eql@line@offset@\relax
3537 \else
3538 \eql@line@offset@
3539 \fi
3540 \eql@arrange@print@aligncenter\eql@line@offset@
3541 \else
3542 \eql@arrange@try\eql@tagwidth@
3543 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3544 \ifdim\eql@line@offset@<\eql@tagwidth@
3545 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3546 \else
3547 \eql@arrange@print@alignright\eql@tagwidth@\z@
3548 \fi
3549 \else
3550 \let\eql@tagpos@reserve\eql@false
3551 \eql@arrange@aligncenter@notag
3552 \fi
3553 \fi
3554 \else
3555 \eql@arrange@aligncenter@notag

```

```

3556 \fi
3557 }

```

10.4 Left Alignment

```

3558 \def\eql@arrange@alignleft@init{%
3559   \eql@tagging@alignleft
3560   \eql@line@offset@dimexpr\eql@marginleft@+\eql@shape@amount@relax
3561   \ifdim\eql@line@offset@<\eql@marginleft@min@
3562     \eql@line@offset@\eql@marginleft@min@
3563   \fi
3564 }

3565 \def\eql@arrange@alignleft@notag{%
3566   \ifdim\eql@line@offset@>\eql@marginleft@min@
3567     \eql@arrange@try\eql@line@offset@
3568     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3569       \eql@arrange@print@alignleft\eql@line@offset@\z@
3570     \else
3571       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3572     \fi
3573   \else
3574     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3575   \fi
3576 }

3577 \def\eql@arrange@alignleft@tagsright{%
3578   \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@relax}%
3579   \ifnum\eql@arrange@badness@<\eql@alignbadness@
3580     \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3581   \else
3582     \ifdim\eql@line@offset@>\eql@marginleft@min@
3583       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@relax}%
3584     \fi
3585     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3586       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3587     \else
3588       \let\eql@tagpos@reserve\eql@false
3589       \eql@arrange@alignleft@notag
3590     \fi
3591   \fi
3592 }

3593 \def\eql@arrange@alignleft@tagsleft{%
3594   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3595     \ifdim\eql@line@offset@>\eql@tagwidth@
3596       \eql@arrange@try\eql@line@offset@
3597       \ifnum\eql@arrange@badness@<\eql@alignbadness@
3598         \eql@arrange@print@alignleft\eql@line@offset@\z@
3599       \else
3600         \eql@arrange@try\eql@tagwidth@
3601         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3602           \eql@arrange@print@alignright\eql@tagwidth@\z@
3603         \else
3604           \let\eql@tagpos@reserve\eql@false
3605           \eql@arrange@print@alignright\eql@marginleft@min@\z@
3606         \fi
3607       \fi
3608   \else
3609     \eql@arrange@try\eql@tagwidth@

```

```

3610     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3611     \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3612     \else
3613     \let\eqL@tagpos@reserve\eqL@false
3614     \eqL@arrange@alignleft@notag
3615     \fi
3616   \fi
3617 \else
3618   \eqL@arrange@alignleft@notag
3619 \fi
3620 }

```

10.5 Right Alignment

```

3621 \def\eqL@arrange@alignright@init{%
3622   \eqL@tagging@alignright
3623   \eqL@line@offset@dimexpr\eqL@marginright@-\eqL@shape@amount@relax
3624   \ifdim\eqL@line@offset@<z@
3625     \eqL@line@offset@z@
3626   \fi
3627 }

```

TODO: describe

```

3628 \def\eqL@arrange@alignright@notag{%
3629   \ifdim\eqL@line@offset@>z@
3630     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3631     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3632       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3633     \else
3634       \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3635     \fi
3636   \else
3637     \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3638   \fi
3639 }

```

TODO: describe

```

3640 \def\eqL@arrange@alignright@tagsright{%
3641   \ifdim\eqL@line@offset@>\eqL@tagwidth@
3642     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3643     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3644       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3645     \else
3646       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3647       \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3648         \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
3649       \else
3650         \let\eqL@tagpos@reserve\eqL@false
3651         \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3652       \fi
3653     \fi
3654   \else
3655     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3656     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3657       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3658     \else
3659       \let\eqL@tagpos@reserve\eqL@false
3660       \eqL@arrange@alignright@notag
3661     \fi

```

```
3662 \fi
3663 }
```

TODO: describe

```
3664 \def\eql@arrange@alignright@tagsleft{%
3665   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3666     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3667     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3668       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3669     \else
3670       \ifdim\eql@line@offset@>\z@
3671         \eql@arrange@try\eql@tagwidth@
3672       \fi
3673       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3674         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3675       \else
3676         \let\eql@tagpos@reserve\eql@false
3677         \eql@arrange@alignright@notag
3678       \fi
3679     \fi
3680   \else
3681     \eql@arrange@alignright@notag
3682   \fi
3683 }
```

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`

```
3684 \def\eql@box@cr{%
3685   \ifmmode\else\unskip\fi
3686   \eql@vspaceskip@\z@skip
3687   \let\eql@punct@term\eql@false
3688   \let\eql@class@rel@composed@empty
3689   \eql@ampprotect\eql@box@cr@test\eql@box@cr@process
3690 }
```

TODO: describe

```
3691 \def\eql@box@cr@test@setopt{\let\eql@box@cr@test\eql@box@cr@testopt}
3692 \def\eql@box@cr@test@setall{\let\eql@box@cr@test\eql@box@cr@testall}
```

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

```
3693 \def\eql@box@cr@testopt#1{\eql@teststaropt@tight
3694   {\eql@box@cr@testopt@set{#1}}{\eql@box@cr@testopt@set{#1}}{0pt}}
3695 \def\eql@box@cr@testopt@set#1[#2]{\advance\eql@vspaceskip@glueexpr#2\relax#1}
```

```

\eql@box@cr@testall TODO: describe
box@cr@testall@parse
3696 \def\eql@box@cr@testall{\eql@parseopt@cr\eql@box@cr@testall@parse}
3697 \def\eql@box@cr@testall@parse{%
3698   \ifx\eql@parseopt@token[%
3699     \let\eql@parseopt@next\eql@parseopt@vspace
3700   \fi
3701   \ifx\eql@parseopt@token*%
3702     \let\eql@parseopt@next\eql@parseopt@gobble
3703   \fi
3704   \ifx\eql@parseopt@token.%
3705     \let\eql@parseopt@next\eql@parseopt@punctpass
3706   \fi
3707   \ifx\eql@parseopt@token,%
3708     \let\eql@parseopt@next\eql@parseopt@punctpass
3709   \fi
3710   \ifx\eql@parseopt@token~%
3711     \let\eql@parseopt@next\eql@parseopt@punctpass
3712   \fi
3713   \ifx\eql@parseopt@token'%
3714     \let\eql@parseopt@next\eql@parseopt@punctnext
3715   \fi
3716   \ifx\eql@parseopt@token!%
3717     \let\eql@parseopt@next\eql@parseopt@punctterm
3718   \fi
3719   \ifx\eql@parseopt@token/%
3720     \let\eql@parseopt@next\eql@parseopt@punctclear
3721   \fi
3722   \ifx\eql@parseopt@token=%
3723     \let\eql@parseopt@next\eql@parseopt@relsyb
3724   \fi
3725   \ifx\eql@parseopt@token;%
3726     \let\eql@parseopt@next\eql@parseopt@relcont
3727   \fi
3728   \ifx\eql@parseopt@token:%
3729     \let\eql@parseopt@next\eql@parseopt@relstart
3730   \fi
3731   \ifx\eql@parseopt@token|%
3732     \let\eql@parseopt@next\eql@box@cr@parse@rel
3733   \fi
3734   \ifx\eql@parseopt@token&%
3735     \let\eql@parseopt@next\eql@parseopt@end
3736   \fi
3737 }
3738 \def\eql@box@cr@parse@rel#1#2{%
3739   \def\eql@tmp{#2}%
3740   \ifx\eql@tmp\eql@relax\else
3741     \eql@punct@next@clear
3742   \fi
3743   \ifx\eql@box@open\eql@box@lines@open
3744     \ifx\eql@tmp\@empty
3745       \def\eql@class@rel@composed{\eql@shape@cont}%
3746     \else
3747       \def\eql@class@rel@composed{\eql@shape@rel#2}%
3748     \fi
3749   \fi
3750   \ifx\eql@box@open\eql@box@columns@open
3751     \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3752   \fi

```

```

3753 \ifx\eql@box@open\eql@box@cases@open
3754 \def\eql@class@rel@composed{&#2}%
3755 \fi
3756 \eql@parseopt@end}

```

\eql@box@cr@process

```

3757 \def\eql@box@cr@process{%
3758 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
3759 \edef\eql@tmp{%
3760 \unexpanded{%
3761 \eql@box@endline
3762 \eql@box@lastcell
3763 \cr
3764 }%
3765 \noalign{%
3766 \vskip\the\eql@vspaceskip@\relax
3767 }%
3768 \unexpanded\expandafter{\eql@class@rel@composed}%
3769 }%
3770 \eql@tmp
3771 }

```

\eql@box@endline

```

3772 \def\eql@box@endline{%
3773 \eql@punct@apply@line
3774 \eql@hook@lineout
3775 }

```

11.2 Column Breaks

TODO: describe

```

3776 \def\eql@box@amp{%
3777 \eql@ampprotecttwo\eql@box@amp@testescape\eql@amp@org
3778 \eql@box@amp@process}
3779 \def\eql@box@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
3780 \relax
3781 \let\eql@punct@term\eql@false
3782 \let\eql@class@rel@composed\@empty
3783 \eql@box@amp@test{#2}}}

```

TODO: describe

```

3784 \def\eql@box@amp@test@setopt{%
3785 \let\eql@box@amp@test\eql@box@amp@testopt}
3786 \def\eql@box@amp@test@setall{%
3787 \let\eql@box@amp@test\eql@box@amp@testall}

```

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

```

3788 \let\eql@box@amp@testopt\@empty

```

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

ox@amp@testall@parse

```

3789 \def\eql@box@amp@testall{\eql@parseopt@cr\eql@box@amp@testall@parse}
3790 \def\eql@box@amp@testall@parse{%

```



```

3791 \ifx\eql@parseopt@token.%
3792   \let\eql@parseopt@next\eql@parseopt@punctpass
3793 \fi
3794 \ifx\eql@parseopt@token,%
3795   \let\eql@parseopt@next\eql@parseopt@punctpass
3796 \fi
3797 \ifx\eql@parseopt@token~%
3798   \let\eql@parseopt@next\eql@parseopt@punctpass
3799 \fi
3800 \ifx\eql@parseopt@token'%
3801   \let\eql@parseopt@next\eql@parseopt@punctnext
3802 \fi
3803 \ifx\eql@parseopt@token!%
3804   \let\eql@parseopt@next\eql@parseopt@punctterm
3805 \fi
3806 \ifx\eql@parseopt@token=%
3807   \let\eql@parseopt@next\eql@parseopt@relsyb
3808 \fi
3809 \ifx\eql@parseopt@token;%
3810   \let\eql@parseopt@next\eql@parseopt@relcont
3811 \fi
3812 \ifx\eql@parseopt@token:%
3813   \let\eql@parseopt@next\eql@parseopt@relstart
3814 \fi
3815 \ifx\eql@parseopt@token|%
3816   \let\eql@parseopt@next\eql@box&@parse@rel
3817 \fi
3818 \ifx\eql@parseopt@token&%
3819   \let\eql@parseopt@next\eql@parseopt@end
3820 \fi
3821 }
3822 \def\eql@box&@parse@rel#1#2{%
3823   \ifx\eql@box@open\eql@box@columns@open
3824     \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3825   \else
3826     \def\eql@class@rel@composed{#2}%
3827   \fi
3828   \eql@parseopt@end}

```

\eql@box&@process

```

3829 \def\eql@box&@process{%
3830   \ifdefined\eql@punct@term\eql@punct@apply@top\fi
3831   \ifx\eql@box@open\eql@box@columns@open
3832     \edef\eql@tmp{%
3833       \ifx\eql@class@rel@composed\@empty
3834         \ifx\eql@box@lastcell\eql@box@columns@lastcell@odd
3835           &\noexpand\eql@punct@next@clear\fi&%
3836       \else
3837         \ifx\eql@box@lastcell\eql@box@columns@lastcell@even&\fi%
3838         \unexpanded\expandafter{\eql@class@rel@composed}%
3839       \fi
3840     }%
3841   \else
3842     \edef\eql@tmp{%
3843       \ifx\eql@class@rel@composed\@empty
3844         &%
3845       \else
3846         \unexpanded\expandafter{\eql@class@rel@composed}%

```

```

3847     \fi
3848   }%
3849 \fi
3850 \eql@tmp
3851 }

```

11.3 Lines Mode

```

3852 \def\eql@box@lines@lastcell{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3853 \def\eql@box@lines@open{%
3854   \eql@shape@align@enable
3855   \let\eql@box@lastcell\eql@box@lines@lastcell
3856   \everycr{\noalign{%
3857     \eql@verbose@info\eql@verbose@msg@startline@number
3858     \global\advance\eql@row@\@ne
3859   }}%
3860   \tabskip\z@skip
3861   \halign\bgroup
3862     &%
3863     \global\let\eql@cell@container\@empty
3864     \setbox\eql@cellbox@\hbox{%
3865       \eql@strut@cell
3866       \@lign
3867       $\m@th\eql@mathstyle
3868       \eql@hook@colin
3869       ##%
3870       \eql@punct@apply@col
3871       \eql@hook@colout
3872       \eql@tagging@mathsave
3873       $%
3874       \eql@tagging@mathaddlast
3875     }%
3876     \ifdefined\eql@shape@lastrow
3877       \eql@totalrows@\eql@row@
3878     \fi
3879     \eql@shape@eval
3880     \eql@cell@container
3881     \ifdefined\eql@frame@cmd
3882       \ifcase\eql@shape@pos@
3883         \eql@frame@measure
3884         \advance\eql@shape@amount@-\eql@frame@margin@
3885       \or\or
3886         \eql@frame@measure
3887         \advance\eql@shape@amount@+\eql@frame@margin@
3888       \fi
3889     \eql@frame@print
3890     \fi
3891     \ifcase\eql@shape@pos@
3892       \kern\eql@shape@amount@
3893       \box\eql@cellbox@
3894       \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3895       -\eql@shape@amount@+\@flushglue\relax
3896     \eql@tagging@alignleft
3897   \or
3898     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3899     \box\eql@cellbox@

```

```

3900      \hskip\glueexpr\eq\paddingright@-\eq\shape@amount@+\@flushglue\relax
3901      \eq\tagging@aligncenter
3902    \or
3903      \hskip\glueexpr\eq\paddingleft@+\eq\paddingright@
3904      +\eq\shape@amount@+\@flushglue\relax
3905      \box\eq\cellbox@
3906      \kern-\eq\shape@amount@
3907      \eq\tagging@alignright
3908    \fi
3909    \tabskip\eq\colsep@\relax
3910  \crrc
3911  \noalign{%
3912    \global\let\eq\shape@lastrow\eq\false
3913    \eq\hook@blockbefore
3914  }%
3915  \eq\hook@blockin
3916 }
3917 \def\eq\box@lines@set{\let\eq\box@open\eq\box@lines@open}

```

11.4 Columns Mode

```

3918 \def\eq\box@columns@lastcell@odd{%
3919   &\omit
3920   \eq\prevwidth@\wd\eq\cellbox@
3921   \let\eq\frame@cmd\eq\frame@prevcmd
3922   \ifdefined\eq\frame@cmd
3923     \eq\frame@measure
3924     \advance\eq\prevwidth@\eq\frame@margin@
3925     \eq\frame@print
3926   \fi
3927   \kern-\eq\prevwidth@
3928   \unhbox\eq\cellbox@
3929   \hfil
3930   &\omit\kern-\eq\colsep@
3931 }%
3932 \def\eq\box@columns@lastcell@even{&\omit\kern-\eq\colsep@}
3933 \def\eq\box@columns@open{%
3934 % \TODO templates
3935   \eq\shape@align@disable
3936   \let\eq\box@lastcell\@empty
3937   \everycr{\noalign{%
3938     \eq\verbose@info\eq\verbose@msg@startline@new
3939   }}%
3940   \tabskip\z@skip
3941   \halign\bgroup
3942     &%
3943     \let\eq\box@lastcell\eq\box@columns@lastcell@odd
3944     \global\let\eq\cell@container\@empty
3945     \global\setbox\eq\cellbox@\hbox{%
3946       \eq\strut@cell
3947       \@lign
3948       $\m@th\eq\mathstyle
3949       \eq\hook@colin
3950       ##%
3951       \eq\punct@apply@next
3952       \eq@class@innerleft
3953       \eq\hook@innerleft
3954       \eq\tagging@mathsave

```

```

3955      $%
3956      \eql@tagging@mathaddlast
3957    }%
3958    \eql@cell@container
3959    \hfil
3960    \kern\wd\eql@cellbox@
3961    \ifdefined\eql@frame@cmd
3962      \eql@frame@measure
3963      \kern\eql@frame@margin@
3964    \fi
3965    \global\let\eql@frame@prevcmd\eql@frame@cmd
3966    \tabskip\z@skip
3967  &%
3968    \eql@prevwidth@\wd\eql@cellbox@
3969    \let\eql@box@lastcell\eql@box@columns@lastcell@even
3970    \let\eql@frame@cmd\eql@frame@prevcmd
3971    \global\let\eql@cell@container\@empty
3972    \setbox\eql@cellbox@\hbox{%
3973      \unhbox\eql@cellbox@
3974      \eql@strut@cell
3975      \@lign
3976      $\m@th\eql@mathstyle
3977      \eql@hook@innerright
3978      \eql@class@innerright@sel
3979      ##%
3980      \eql@punct@apply@col
3981      \eql@hook@colout
3982      \eql@tagging@mathsave
3983      $%
3984      \eql@tagging@mathaddlast
3985    }%
3986    \eql@cell@container
3987    \ifdefined\eql@frame@cmd
3988      \eql@frame@measure
3989      \advance\eql@prevwidth@\eql@frame@margin@
3990      \eql@frame@print
3991    \fi
3992    \kern-\eql@prevwidth@
3993    \unhbox\eql@cellbox@
3994    \hfil
3995    \tabskip\eql@colsep@\relax
3996  \crr
3997  \noalign{%
3998    \eql@hook@blockbefore
3999  }%
4000  \eql@hook@blockin
4001 }
4002 \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?

```

4003 \def\eql@box@cases@lastcell{&}%
4004 \let\eql@box@cases@condtext\eql@false

```

```

4005 \let\eql@box@cases@condintro\@empty

4006 \def\eql@box@cases@open{%
4007   \eql@shape@align@disable
4008   \let\eql@box@lastcell\@empty
4009   \everycr{\noalign{%
4010     \eql@verbose@info\eql@verbose@msg@startline@new
4011   }}%
4012   \tabskip\z@skip
4013   \halign\bgroup
4014     \let\eql@box@lastcell\eql@box@cases@lastcell
4015     \global\let\eql@cell@container\@empty
4016     \global\setbox\eql@cellbox@\hbox{%
4017       \eql@strut@cell
4018       \@lign
4019       $\m@th\eql@mathstyle
4020       \eql@hook@colin
4021       ##%
4022       \eql@punct@apply@next
4023       \eql@tagging@mathsave
4024       $%
4025       \eql@tagging@mathaddlast
4026     }%
4027     \eql@cell@container
4028     \unhbox\eql@cellbox@
4029     \hfil
4030     \eql@tagging@alignleft
4031     \tabskip\eql@colsep@\relax
4032   &%
4033   \let\eql@box@lastcell\@empty
4034   \global\let\eql@cell@container\@empty
4035   \setbox\eql@cellbox@\hbox{%
4036     \unhbox\eql@cellbox@
4037     \eql@strut@cell
4038     \@lign
4039     $\m@th\eql@mathstyle
4040     \ifdefined\eql@box@cases@condtext
4041       \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
4042     \eql@box@cases@condintro
4043     ##%
4044     \eql@punct@apply@col
4045     \egroup
4046     \eql@hook@colout
4047     \eql@tagging@mathsave
4048     $%
4049     \eql@tagging@mathaddlast
4050   }%
4051   \eql@cell@container
4052   \unhbox\eql@cellbox@
4053   \hfil
4054   \eql@tagging@alignleft
4055   \tabskip\z@skip
4056 \crrc
4057 \noalign{%
4058   \eql@hook@blockbefore
4059 }%
4060 \eql@hook@blockin
4061 }

```

```

4062 \def\eql@box@cases@set{%
4063   \ifdefined\eql@cases@mathstyle\let\eql@mathstyle\eql@cases@mathstyle\fi
4064   \let\eql@box@open\eql@box@cases@open}

```

11.6 Main

```

4065 \let\eql@box@box\vcenter
4066 \let\eql@box@open\@undefined
4067 \let\eql@box@frame\@firstofone
4068 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

4069 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4070   \left#1##1\right#2}}
4071 \def\eql@box@getdim{\setbox\@ne\hbox{\null\ht\@ne\ht\z@\dp\@ne\dp\z@}
4072 \def\eql@box@deldim#1{\hbox{$\m@th\hbox{\null\delimiterspace\z@\left#1
4073   \ifx\eql@box@box\vcenter\vcenter{\box\@ne}\else\box\@ne\fi\right.$}}
4074 \def\eql@box@ldelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4075   \eql@box@getdim\eql@box@deldim#1##1}}
4076 \def\eql@box@rdelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4077   \eql@box@getdim##1\eql@box@deldim#1}}

```

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

```

4078 \def\eql@box@close{%
4079   \ifvmode\else
4080     \ifmmode\else\unskip\fi
4081     \global\let\eql@shape@lastrow\eql@true
4082     \eql@punct@apply@block
4083     \ifdefined\eql@box@punct@term
4084       \eql@punct@apply@top
4085     \fi
4086     \eql@box@endline
4087     \eql@box@lastcell
4088     \cr
4089   \fi
4090   \noalign{%
4091     \eql@hook@blockafter
4092     \global\let\eql@shape@lastrow\eql@false
4093   }%
4094   \eql@tagging@tablesaveinner
4095 \egroup
4096 }

```

`\eql@box@vcenter`

```

4097 \def\eql@box@vcenter#1{%
4098   \ifmmode
4099     \vcenter{#1}%
4100   \else
4101     $\m@th\vcenter{#1}$%
4102   \fi
4103 }

```

`\eql@box@start`

```

4104 \let\eql@box@endmath\eql@false

```

```

4105 \def\eql@box@start{%
4106   \relax
4107   \ifmmode
4108     \let\eql@box@endmath\eql@false
4109   \else
4110     \let\eql@box@endmath\eql@true
4111     \expandafter$%$
4112   \fi
4113   \eql@stack@save@box
4114   \let\eql@frame@cmd\@undefined
4115   \let\eql@layoutleft\eql@false
4116   \eql@row@z@
4117   \eql@totalrows@\@M
4118   \eql@shape@select
4119   \setbox\z@\ifx\eql@box@box\vcenter
4120     \expandafter\vbox
4121   \else
4122     \expandafter\eql@box@box
4123   \fi\bgroup
4124   \let\eqnpunct\eql@punct@setnext
4125   \eql@display@nest
4126   \let\\eql@box@cr
4127   \ifdefined\eql@amp@mode
4128     \let\&\eql@box@amp
4129   \fi
4130   \eql@spread@set
4131   \eql@strut@make
4132   \eql@box@open
4133 }

```

\eql@box@end

```

4134 \def\eql@box@end{%
4135   \eql@box@close
4136   \egroup
4137   \eql@box@frame{%
4138     \ifdefined\eql@display@marginleft
4139       \hskip\glueexpr\eql@display@marginleft\relax
4140     \fi
4141     \ifx\eql@box@box\vcenter
4142       \eql@box@vcenter{\unvbox\z@}%
4143     \else
4144       \box\z@
4145     \fi
4146     \eql@tagging@tableaddinner
4147     \ifdefined\eql@display@marginright
4148       \hskip\glueexpr\eql@display@marginright\relax
4149     \fi
4150   }%
4151   \eql@stack@restore
4152   \ifdefined\eql@box@endmath
4153     \expandafter$%$
4154   \fi
4155 }

```

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

```

4156 \def\eql@box@main{%
4157   \eql@box@start

```

```

4158 \eql@scan@body
4159 \eql@box@end
4160 }

```

11.7 Options Processing

TODO: describe

```

4161 \def\eql@box@test@setopt{\let\eql@box@test\eql@box@testopt}
4162 \def\eql@box@test@setall{\let\eql@box@test\eql@box@testall}

```

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

```

4163 \def\eql@box@testopt#1{\eql@testopt@tight{\eql@box@testopt@set{#1}}{}}
4164 \def\eql@box@testopt@set#1[#2]{\eqnaddopt{#2}#1}

4165 \def\eql@box@testall{\eql@parseopt@env\eql@box@testall@parse}
4166 \def\eql@box@testall@parse{%
4167   \ifx\eql@parseopt@token[%
4168     \let\eql@parseopt@next\eql@parseopt@opt
4169   \fi
4170   \ifx\eql@parseopt@token.%
4171     \let\eql@parseopt@next\eql@parseopt@punctpass
4172   \fi
4173   \ifx\eql@parseopt@token,%
4174     \let\eql@parseopt@next\eql@parseopt@punctpass
4175   \fi
4176   \ifx\eql@parseopt@token~%
4177     \let\eql@parseopt@next\eql@parseopt@punctpass
4178   \fi
4179   \ifx\eql@parseopt@token'
4180     \let\eql@parseopt@next\eql@parseopt@punctopt
4181   \fi
4182   \ifx\eql@parseopt@token!%
4183     \let\eql@parseopt@next\eql@box@parse@punctterm
4184   \fi
4185   \ifx\eql@parseopt@token=%
4186     \let\eql@parseopt@next\eql@parseopt@lines
4187   \fi
4188   \ifx\eql@parseopt@token|%
4189     \let\eql@parseopt@next\eql@parseopt@columns
4190   \fi
4191   \ifx\eql@parseopt@token<%
4192     \let\eql@parseopt@next\eql@parseopt@ampeq
4193   \fi
4194   \ifx\eql@parseopt@token>%
4195     \let\eql@parseopt@next\eql@parseopt@eqamp
4196   \fi
4197 }
4198 \def\eql@box@parse@punctterm#1{\eqnaddopt{punctterm}\eql@parseopt@peek}

```

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

```

4199 \def\eql@box@end@testall{\eql@parseopt@env\eql@box@end@testall@parse}
4200 \def\eql@box@end@testall@parse{%
4201   \ifx\eql@parseopt@token.%
4202     \let\eql@parseopt@next\eql@parseopt@punctpass
4203   \fi

```



```

4204 \ifx\eql@parseopt@token,%
4205   \let\eql@parseopt@next\eql@parseopt@punctpass
4206 \fi
4207 \ifx\eql@parseopt@token~%
4208   \let\eql@parseopt@next\eql@parseopt@punctpass
4209 \fi
4210 \ifx\eql@parseopt@token'%
4211   \let\eql@parseopt@next\eql@parseopt@punctblock
4212 \fi
4213 \ifx\eql@parseopt@token!%
4214   \let\eql@parseopt@next\eql@box@end@parse@punctterm
4215 \fi
4216 }
4217 \def\eql@box@end@parse@punctterm#1{%
4218   \let\eql@box@punct@term\eql@true\eql@parseopt@peek}

```

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

```

4219 \def\eql@box@processopt{%
4220   \let\eql@box@frame\@firstofone
4221   \let\eql@display@marginleft\@undefined
4222   \let\eql@display@marginright\@undefined
4223   \let\eql@box@punct@term\eql@false
4224   \let\eql@punct@block\@undefined
4225   \eql@nextopt@process{box}%
4226   \let\eql@punct@next\@undefined
4227   \eql@colsep@\glueexpr\eql@box@colsep\relax
4228   \ifdefined\eql@paddingleft@val
4229     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
4230   \else
4231     \eql@paddingleft@\z@
4232   \fi
4233   \ifdefined\eql@paddingright@val
4234     \eql@paddingright@\glueexpr\eql@paddingright@val\relax
4235   \else
4236     \eql@paddingright@\z@
4237   \fi
4238   \eql@indent@\glueexpr\eql@indent@val\relax
4239 }

```

11.8 Environment

equationsbox (*env.*)

```

4240 \newenvironment{equationsbox}{%
4241   \eql@verbose@info\eql@verbose@msg@enterenv
4242   \ifdefined\eql@box@env@modifier
4243     \eql@box@test@setall
4244   \else
4245     \eql@box@nomodifier
4246   \fi
4247   \eql@ampprotect\eql@box@test\eql@box@env@start
4248 }{%
4249   \ifdefined\eql@box@doscan\else
4250     \expandafter\eql@box@end
4251   \fi
4252   \eql@verbose@info\eql@verbose@msg@leaveenv
4253 }

```

`\eql@box@env@start`

```
4254 \def\eql@box@env@start{%
4255   \eql@box@processopt
4256   \ifdefined\eql@box@doscan
4257     \eql@box@call@set
4258     \expandafter\eql@scan@env
4259   \else
4260     \expandafter\eql@box@start
4261   \fi
4262 }
```

`\eql@box@call`

```
4263 \def\eql@box@call{\eql@box@main\eql@scan@end}
4264 \def\eql@box@call@test{%
4265   \eql@ampprotect\eql@box@end@testall\eql@box@call}
4266 \def\eql@box@call@set{%
4267   \ifdefined\eql@box@end@modifier
4268     \let\eql@scan@call\eql@box@call@test
4269   \else
4270     \let\eql@scan@call\eql@box@call
4271   \fi
4272 }
```

`\eql@box@ang@open`

```
4273 \newenvironment{equationsbox@ang}{}{}
4274 \def\eql@box@ang@open{%
4275   \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4276   \begin{equationsbox@ang}%
4277   \eql@verbose@info\eql@verbose@msg@enterenv
4278   \let\>\eql@box@ang@close
4279   \ifdefined\eql@box@ang@modifier
4280     \eql@box@test@setall
4281   \else
4282     \eql@box@nomodifier
4283   \fi
4284   \eql@ampprotect\eql@box@test\eql@box@ang@start
4285 }
```

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

```
4286 \def\eql@box@ang@start{%
4287   \eql@box@processopt
4288   \ifdefined\eql@box@doscan
4289     \eql@box@call@set
4290     \expandafter\eql@scan@ang
4291   \else
4292     \expandafter\eql@box@start
4293   \fi
4294 }
```

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

```
4295 \def\eql@box@ang@close{%
4296   \ifdefined\eql@box@doscan
4297     \let\eql@box@end@modifier\eql@false
4298   \fi
```

```

4299 \ifdefined\eql@box@end@modifier
4300 \expandafter\eql@ampprotect\expandafter\eql@box@end@testall
4301 \fi
4302 \eql@box@ang@end
4303 }

```

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

```

4304 \def\eql@box@ang@end{%
4305 \ifdefined\eql@box@doscans\else
4306 \expandafter\eql@box@end
4307 \fi
4308 \eql@verbose@info\eql@verbose@msg@leaveenv
4309 \end{equationsbox@ang}%
4310 \ignorespaces
4311 }

```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

4312 \def\eql@single@start@native{%
4313 \eql@display@init
4314 \eql@display@print
4315 \let\raisetag\eql@raisetag@default
4316 \eql@shape@align@disable
4317 \eql@hook@eqin
4318 % \mathopen{}}%
4319 }%

```

TODO: describe

```

4320 \def\eql@single@end@native{%
4321 % \mathclose{}}%
4322 \eql@tags@container
4323 \eql@numbering@single@eval
4324 \if@eqnsw
4325 \ifdefined\eql@tagsleft
4326 \leqno
4327 \else
4328 \eqno
4329 \fi
4330 \eql@composetag@print
4331 \fi
4332 \eql@interline@container
4333 \advance\eql@belowspace@\eql@vspaceskip@
4334 \eql@display@container
4335 \eql@display@penalty
4336 \eql@display@vspace@native
4337 }%

```

12.2 Print

```

4338 \def\eql@single@start@print{%
4339 \eql@display@init
4340 \eql@display@print

```

```

4341 \eqL@shape@align@enable
4342 \eqL@totalrows@ \@ne
4343 \eqL@row@ \@ne
4344 \eqL@arrange@init
4345 \global\let\eqL@cell@container\@empty
4346 \prevgraf\numexpr\prevgraf+\@ne\relax
4347 \setbox\eqL@cellbox@\hbox\bgroup
4348   \eqL@restore@hfuzz
4349   \eqL@strut@cell
4350   $\m@th\eqL@mathstyle%$
4351   \eqL@hook@eqin
4352 }
4353 \def\eqL@single@end@print{%
4354   \eqL@tagging@mathsave
4355   $%$
4356   \hfil
4357   \kern\z@
4358   \egroup
4359   \prevgraf\numexpr\prevgraf-\@ne\relax
4360   \eqL@shape@eval
4361   \eqL@cell@container
4362   \ifdefined\eqL@frame@cmd
4363     \eqL@frame@adjust
4364   \fi
4365   \eqL@cellwidth@\wd\eqL@cellbox@
4366   \eqL@line@height@\ht\eqL@cellbox@
4367   \eqL@line@depth@\dp\eqL@cellbox@
4368   \eqL@totalwidth@\eqL@cellwidth@
4369   \eqL@totalheight@\dimexpr\eqL@line@height@+\eqL@line@depth@\relax
4370   \eqL@topheight@\eqL@line@height@
4371   \eqL@bottomdepth@\eqL@line@depth@
4372   \eqL@tags@container
4373   \eqL@numbering@single@eval
4374   \if@eqnsw
4375     \eqL@tagbox@make\eqL@composetag@print
4376     \eqL@tagrows@\@ne
4377     \ifdefined\eqL@tagpos@reserve\else
4378       \eqL@tagwidth@\z@
4379     \fi
4380     \eqL@tagheight@block@\ht\eqL@tagbox@
4381     \eqL@tagdepth@block@\dp\eqL@tagbox@
4382   \else
4383     \eqL@numbering@warnunused
4384     \eqL@tagwidth@\z@
4385     \eqL@tagrows@\z@
4386   \fi
4387   \eqL@tagwidth@max@\eqL@tagwidth@
4388   \eqL@tagpos@single@eval
4389   \eqL@tagpos@print@line@eval
4390   \eqL@intercolumns@\z@
4391   \eqL@adjust@calc@lines
4392   \eqL@display@halign@init{ }%
4393   \halign{##\crr
4394     \noalign{\eqL@display@halign@start}%
4395     \eqL@arrange@print@line

```

```

4396 \cr
4397 \noalign{\eqldisplay@halign@end}%
4398 \eqldtagging@tablesavelines
4399 }%
4400 \eqldtagpos@print@line@end
4401 \eqldisplay@close
4402 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```

4403 \def\eqldlines@measure@line@begin{%
4404 \eqldverbose@info\eqldverbose@msg@startline@number
4405 \eqldnumbering@measure@line@begin
4406 \eqldhook@linein
4407 }

```

TODO: describe

```

4408 \def\eqldlines@measure@line@end{%
4409 \eqldpunct@apply@line
4410 \eqldhook@lineout
4411 }

```

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

```

4412 \def\eqldlines@measure@cell{%
4413 \ifdefined\eqldframe@cmd
4414 \ifcase\eqldshape@pos@
4415 \eqldframe@measure
4416 \advance\eqldshape@amount@-\eqldframe@margin@
4417 \or\or
4418 \eqldframe@measure
4419 \advance\eqldshape@amount@+\eqldframe@margin@
4420 \fi
4421 \eqldframe@print
4422 \fi
4423 \eqldcellwidth@\wd\eqldcellbox@
4424 \eqldline@height@\ht\eqldcellbox@
4425 \eqldline@depth@\dp\eqldcellbox@
4426 \eqlddimensions@startrow
4427 \eqlddimensions@savercell
4428 \kern\eqldcellwidth@
4429 }

```

`\eqldlines@measure`

```

4430 \def\eqldlines@measure{%
4431 \eqldverbose@infoarg\eqldverbose@msg@enter\eqldlines@measure
4432 \eqldmeasure@init\eqldlines@measure@line@begin\eqldlines@measure@line@end
4433 \ifdefined\eqldamp@mode
4434 \let&\eqldbreak@amp

```

```

4435 \fi
4436 \eq@totalrows@\@M
4437 \eq@shape@select

4438 \setbox\z@\vbox{\measuring@true\halign{%
4439   \global\let\eq@cell@container\@empty
4440   \setbox\eq@cellbox@\hbox{%
4441     \eq@strut@cell
4442     \@lign
4443     $\m@th\eq@mathstyle
4444     \eq@hook@colin
4445     ##%
4446     \eq@punct@apply@col
4447     \eq@hook@colout
4448     $%
4449   }%
4450   \ifdefined\eq@shape@lastrow
4451     \eq@totalrows@\eq@row@
4452   \fi
4453   \eq@shape@eval
4454   \eq@cell@container
4455   \eq@lines@measure@cell
4456   \eq@measure@tag
4457   \eq@measure@endrow
4458 \crr

4459 \noalign{%
4460   \global\let\eq@shape@lastrow\eq@false
4461   \eq@hook@blockbefore
4462 }%
4463 \eq@hook@blockin
4464 \eq@scan@body
4465 \ifvmode\else
4466   \global\let\eq@shape@lastrow\eq@true
4467   \eq@punct@apply@block
4468   \eq@hook@blockout
4469   \eq@multi@endline
4470   \cr
4471 \fi
4472 \omit
4473 \cr
4474 \noalign{%
4475   \eq@hook@blockafter
4476   \global\let\eq@shape@lastrow\eq@false
4477 }%
4478 }%

4479 \eq@measure@close

4480 \setbox\z@\vbox{%
4481   \unvbox\z@
4482   \unpenalty
4483   \global\setbox\@ne\lastbox
4484 }%
4485 \eq@totalwidth@\wd\@ne

4486 \eq@verbose@infoarg\eq@verbose@msg@leave\eq@lines@measure
4487 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```
4488 \def\eql@lines@adjust{%
4489   \eql@tagpos@adjust@eval
4490   \eql@adjust@calc@lines
4491   \eql@numbering@best@eval
4492 }
```

13.3 Print

TODO: describe

mes@print@line@begin

```
4493 \def\eql@lines@print@line@begin{%
4494   \eql@verbose@info\eql@verbose@msg@startline@number
4495   \eql@numbering@print@line@begin
4496   \eql@hook@linein
4497 }
```

TODO: describe

```
4498 \def\eql@lines@print@line@end{%
4499   \eql@punct@apply@line
4500   \eql@hook@lineout
4501 }
```

TODO: describe

```
4502 \def\eql@lines@print@line@adjust{%
4503   \ifdefined\eql@frame@cmd
4504     \ifcase\eql@shape@pos@
4505       \eql@frame@measure
4506       \advance\eql@shape@amount@-\eql@frame@margin@
4507     \or\or
4508       \eql@frame@measure
4509       \advance\eql@shape@amount@+\eql@frame@margin@
4510     \fi
4511     \eql@frame@adjust
4512 \fi
4513 \eql@cellwidth@\wd\eql@cellbox@
4514 \eql@line@height@\ht\eql@cellbox@
4515 \eql@line@depth@\dp\eql@cellbox@
4516 \eql@numbering@print@line@eval
4517 \if@eqnsw
4518   \eql@tagbox@make\eql@composetag@print
4519 \fi
4520 \eql@tagpos@print@line@eval
4521 \eql@arrange@print@line
4522 \eql@tagpos@print@line@end
4523 }
```

TODO: describe

```
4524 \def\eql@lines@print{%
4525   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@print
4526   \eql@arrange@init
4527   \eql@display@halign@init\eql@lines@print@line@begin
```

```

4528 \eql@multi@cr@let\eql@lines@print@line@end
4529 \ifdefined\eql@amp@mode
4530 \let\&\eql@break@amp
4531 \fi
4532 \tabskip\z@skip

4533 \halign{%
4534 \global\let\eql@cell@container\@empty
4535 \setbox\eql@cellbox@\hbox{%
4536 \eql@restore@hfuzz
4537 \eql@strut@cell
4538 \@lign
4539 $\m@th\eql@mathstyle
4540 \eql@hook@colin
4541 ##%
4542 \eql@punct@apply@col
4543 \eql@hook@colout
4544 \eql@tagging@mathsave
4545 $%
4546 \hfil
4547 \kern\z@
4548 }%
4549 \eql@shape@eval
4550 \eql@cell@container
4551 \eql@lines@print@line@adjust
4552 \crrr

4553 \noalign{%
4554 \eql@display@halign@start
4555 \eql@numbering@print@block@begin
4556 \eql@hook@blockbefore
4557 }%
4558 \eql@hook@blockin
4559 \eql@scan@body
4560 \ifvmode\else
4561 \relax
4562 \eql@punct@apply@block
4563 \eql@hook@blockout
4564 \eql@multi@endline
4565 \cr
4566 \fi
4567 \noalign{%
4568 \eql@hook@blockafter
4569 \eql@display@halign@end
4570 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@print
4571 }%
4572 \eql@tagging@tablesavelines
4573 }%
4574 }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```
\eql@columns@add@amp
@columns@completerow
4575 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
4576 \def\eql@columns@completerow{%
4577   \count@=\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
4578   \edef\eql@tmp{%
4579     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
4580   \eql@tmp
4581 }

4582 \def\eql@columns@overfull{%
4583   \dimen@=\eql@line@width@
4584   \advance\dimen@-\hfuzz
4585   \ifdim\dimen@>\displaywidth
4586     \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
4587     \wd\z@\z@
4588     \ht\z@=\eql@line@height@
4589     \dp\z@=\eql@line@depth@
4590     \box\z@
4591   \fi
4592 }
```

14.2 Column Breaks

TODO: describe

TODO: describe

```
4593 \let\eql@amp@org\&

4594 \def\eql@columns@amp{%
4595   \eql@ampprotecttwo\eql@columns@amp@testescape\eql@amp@org
4596   \eql@columns@amp@process}
4597 \def\eql@columns@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
4598   \relax
4599   \let\eql@punct@term\eql@false
4600   \let\eql@class@rel@composed\@empty
4601   \eql@columns@amp@test{#2}}}
```

TODO: describe

```
4602 \def\eql@columns@amp@test@setopt{%
4603   \let\eql@columns@amp@test\eql@columns@amp@testopt}
4604 \def\eql@columns@amp@test@setall{%
4605   \let\eql@columns@amp@test\eql@columns@amp@testall}
```

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

```
4606 \let\eql@columns@amp@testopt\@empty
```

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

```
ns@amp@testall@parse
4607 \def\eql@columns@amp@testall{\eql@parseopt@cr\eql@columns@amp@testall@parse}
4608 \def\eql@columns@amp@testall@parse{%
4609   \ifx\eql@parseopt@token.%
```

```

4610 \let\eql@parseopt@next\eql@parseopt@punctpass
4611 \fi
4612 \ifx\eql@parseopt@token,%
4613 \let\eql@parseopt@next\eql@parseopt@punctpass
4614 \fi
4615 \ifx\eql@parseopt@token~%
4616 \let\eql@parseopt@next\eql@parseopt@punctpass
4617 \fi
4618 \ifx\eql@parseopt@token'%
4619 \let\eql@parseopt@next\eql@parseopt@punctnext
4620 \fi
4621 \ifx\eql@parseopt@token!%
4622 \let\eql@parseopt@next\eql@parseopt@punctterm
4623 \fi
4624 \ifx\eql@parseopt@token=%
4625 \let\eql@parseopt@next\eql@parseopt@relsybm
4626 \fi
4627 \ifx\eql@parseopt@token;%
4628 \let\eql@parseopt@next\eql@parseopt@relcont
4629 \fi
4630 \ifx\eql@parseopt@token:%
4631 \let\eql@parseopt@next\eql@parseopt@relstart
4632 \fi
4633 \ifx\eql@parseopt@token|%
4634 \let\eql@parseopt@next\eql@columns@amp@parse@rel
4635 \fi
4636 \ifx\eql@parseopt@token&%
4637 \let\eql@parseopt@next\eql@parseopt@end
4638 \fi
4639 }
4640 \def\eql@columns@amp@parse@rel#1#2{%
4641 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
4642 \eql@parseopt@end}

```

@columns@amp@process

```

4643 \def\eql@columns@amp@process{%
4644 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
4645 \edef\eql@tmp{%
4646 \ifx\eql@class@rel@composed\@empty
4647 \ifodd\eql@column@&\noexpand\eql@punct@next@clear\fi&%
4648 \else
4649 \ifodd\eql@column@&\else&\fi%
4650 \unexpanded\expandafter{\eql@class@rel@composed}%
4651 \fi
4652 }%
4653 \eql@tmp
4654 }

```

14.3 Transpose

TODO: describe

TODO: adjust to \&?!

TODO: describe

```

4655 \let\eql@transpose@active\eql@false
4656 \def\eql@transpose@end{\eql@transpose@end}

```

```

4657 \def\eql@transpose@skip{&\eql@punct@next@clear}
4658 \def\eql@transpose@complete{%
4659   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

TODO: describe

```

4660 \def\eql@transpose{%
4661   \eql@totalcolumns@z@
4662   \eql@totalrows@z@
4663   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
4664   \eql@scan@reg@{}%
4665   \eql@row@z@
4666   \eql@transpose@output@row
4667 }

```

TODO: describe

```

4668 \def\eql@transpose@save@col#1{%
4669   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
4670     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

TODO: describe

```

4671 \def\eql@transpose@scan@col#1&{%
4672   \def\eql@tmpa{#1}%
4673   \ifx\eql@tmpa\eql@transpose@end\else
4674     \advance\eql@totalcolumns@ \@ne
4675     \eql@row@z@
4676     \let\eql@transpose@data@col@\empty
4677     \eql@transpose@scan@row#1\\ \eql@transpose@end\\
4678     \ifnum\eql@row@>\eql@totalrows@
4679       \eql@totalrows@\eql@row@
4680     \fi
4681     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
4682     \expandafter\eql@transpose@scan@col
4683   \fi
4684 }

```

TODO: describe

```

4685 \def\eql@transpose@append@row#1{%
4686   \advance\eql@row@\@ne
4687   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}

```

TODO: describe

```

4688 \def\eql@transpose@scan@row#1\\{%
4689   \def\eql@tmpa{#1}%
4690   \ifx\eql@tmpa\eql@transpose@end\else
4691     \ifx\eql@transpose@active+
4692       \eql@transpose@scan@cell#1&\eql@transpose@end&%
4693     \else
4694       \eql@transpose@append@row{#1}%
4695     \fi
4696     \expandafter\eql@transpose@scan@row
4697   \fi
4698 }

```

TODO: describe

```

4699 \def\eql@transpose@scan@cell#1&#2&{%
4700   \def\eql@tmpa{#2}%

```

```

4701 \ifx\eql@tmpa\eql@transpose@end
4702 \eql@transpose@append@row{#1}%
4703 \else
4704 \eql@transpose@append@row{#1#2}%
4705 \expandafter\eql@transpose@scan@cell@next
4706 \fi
4707 }

```

TODO: describe

```

4708 \def\eql@transpose@scan@cell@next#1{%
4709 \def\eql@tmpa{#1}%
4710 \ifx\eql@tmpa\eql@transpose@end\else
4711 \eql@transpose@append@row{&#1}%
4712 \expandafter\eql@transpose@scan@cell@next
4713 \fi
4714 }

```

TODO: describe

```

4715 \def\eql@transpose@output@row{%
4716 \ifnum\eql@row@<\eql@totalrows@
4717 \advance\eql@row@\@ne
4718 \eql@column@\z@
4719 \eql@transpose@output@col
4720 \ifnum\eql@row@<\eql@totalrows@
4721 \eql@scan@addto\\%
4722 \fi
4723 \expandafter\eql@transpose@output@row
4724 \fi
4725 }

```

TODO: describe

```

4726 \def\eql@transpose@output@col{%
4727 \ifnum\eql@column@<\eql@totalcolumns@
4728 \advance\eql@column@\@ne
4729 \csname eql@transpose@data@col@\the\eql@column@\endcsname
4730 \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4731 \ifnum\eql@column@<\eql@totalcolumns@
4732 \eql@scan@addto{\eql@transpose@complete}%
4733 \fi
4734 \expandafter\eql@transpose@output@col
4735 \fi
4736 }

```

14.4 Measure

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

s@measure@line@begin

```

4737 \def\eql@columns@measure@line@begin{%
4738 \eql@verbose@info\eql@verbose@msg@startline@number
4739 \global\eql@column@\z@
4740 \global\eql@line@height@\z@
4741 \global\eql@line@depth@\z@
4742 \eql@numbering@measure@line@begin
4743 \eql@hook@linein
4744 }

```

```

4745 \def\eql@columns@measure@cell{%
4746   \eql@cellwidth@\wd\eql@cellbox@
4747   \ifdefined\eql@frame@cmd
4748     \eql@frame@measure
4749     \advance\eql@cellwidth@\eql@frame@margin@
4750   \fi
4751   \ifdim\ht\eql@cellbox@>\eql@line@height@
4752     \global\eql@line@height@\ht\eql@cellbox@
4753   \fi
4754   \ifdim\dp\eql@cellbox@>\eql@line@depth@
4755     \global\eql@line@depth@\dp\eql@cellbox@
4756   \fi
4757   \ifnum\eql@column@=\@ne
4758     \eql@dimensions@startrow
4759   \fi
4760   \ifodd\eql@column@
4761     \eql@shape@pos@\tw@
4762   \else
4763     \eql@shape@pos@\z@
4764   \fi
4765   \eql@shape@amount@\z@
4766   \eql@dimensions@savecell
4767   \ifodd\eql@column@\else
4768     \eql@dimensions@savesep
4769   \fi
4770   \kern\eql@cellwidth@
4771 }

```

ms@measure@line@end

```

4772 \def\eql@columns@measure@line@end{%
4773   \eql@punct@apply@line
4774   \eql@hook@lineout
4775   &\omit
4776   \ifnum\eql@column@>\eql@totalcolumns@
4777     \global\eql@totalcolumns@\eql@column@
4778   \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

```

4779   \ifdefined\eql@frame@cmd
4780     \advance\eql@column@\@ne
4781     \wd\eql@cellbox@\z@
4782   \eql@columns@measure@cell
4783   \fi
4784   \eql@measure@tag
4785   \eql@measure@endrow
4786 }

```

\eql@columns@measure

```

4787 \def\eql@columns@measure{%
4788   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@measure
4789   \eql@totalcolumns@\z@
4790   \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end
4791   \ifdefined\eql@amp@mode
4792     \let&\eql@columns@amp
4793   \fi

```

```

4794 \setbox\z@\vbox{\measuring@true\halign{%
4795   &%
4796   \global\advance\eql@column@\@ne
4797   \global\let\eql@cell@container\@empty
4798   \global\setbox\eql@cellbox@\hbox{%
4799     \eql@strut@cell
4800     \@lign
4801     $\m@th\eql@mathstyle
4802     \eql@hook@colin
4803     ##%
4804     \eql@punct@apply@next
4805     \eql@class@innerleft
4806     \eql@hook@innerleft
4807     $%
4808   }%
4809   \eql@cell@container
4810   \hfil
4811   \eql@columns@measure@cell
4812   \global\let\eql@frame@prevcmd\eql@frame@cmd
4813   &%
4814   \eql@prevwidth@\wd\eql@cellbox@
4815   \let\eql@frame@cmd\eql@frame@prevcmd
4816   \global\advance\eql@column@\@ne
4817   \global\let\eql@cell@container\@empty
4818   \setbox\eql@cellbox@\hbox{%
4819     \eql@strut@cell
4820     \@lign
4821     $\m@th\eql@mathstyle
4822     \eql@hook@innerright
4823     \eql@class@innerright@sel
4824     ##%
4825     \eql@punct@apply@col
4826     \eql@hook@colout
4827     $%
4828   }%
4829   \eql@cell@container
4830   \eql@columns@measure@cell
4831   \hfil
4832 \crrr

4833 \noalign{%
4834   \eql@hook@blockbefore
4835 }%
4836 \eql@hook@blockin
4837 \eql@scan@body

4838 \ifvmode\else
4839   \eql@punct@apply@block
4840   \eql@hook@blockout
4841   \eql@multi@endline
4842   \cr
4843 \fi
4844 \noalign{%
4845   \eql@hook@blockafter
4846 }%

```

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

```

4847 \omit

```

```

4848 \eql@column@{\@ne
4849 \eql@columns@completerow
4850 \cr
4851 }%

4852 \eql@measure@close

4853 \setbox\z@\vbox{%
4854 \unvbox\z@
4855 \unpenalty
4856 \global\setbox\@ne\lastbox
4857 }%
4858 \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4859 \let\eql@colwidth@tab\@empty
4860 \loop
4861 \setbox\@ne\hbox{%
4862 \unhbox\@ne
4863 \unskip
4864 \global\setbox\thr@@\lastbox
4865 }%
4866 \ifhbox\thr@@
4867 \eql@colwidth@save{\wd\thr@@}%
4868 \repeat

4869 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@measure
4870 }

```

14.5 Columns Placement

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

```

4871 \def\eql@columns@adjust{%
4872 \ifodd\eql@totalcolumns@
4873 \advance\eql@totalcolumns@\@ne
4874 \fi
4875 \eql@tagpos@adjust@eval
4876 \eql@adjust@calc@columns
4877 }

```

14.6 Print

TODO: describe

mns@print@line@begin

```

4878 \def\eql@columns@print@line@begin{%
4879 \eql@verbose@info\eql@verbose@msg@startline@number
4880 \global\eql@column@\z@
4881 \global\eql@line@pos@\eql@marginleft@
4882 \global\eql@line@width@\z@
4883 \global\eql@line@avail@\eql@totalwidth@
4884 \global\eql@line@height@\z@
4885 \global\eql@line@depth@\z@
4886 \eql@numbering@print@line@begin

```

```

4887 \eql@hook@linein
4888 }

```

l@columns@print@cell

```

4889 \def\eql@columns@print@cell{%
4890 \eql@cellwidth@\wd\eql@cellbox@
4891 \ifodd\eql@column@
4892 \ifdefined\eql@frame@cmd
4893 \eql@frame@measure
4894 \advance\eql@cellwidth@\eql@frame@margin@
4895 \fi
4896 \dimen@z@
4897 \else
4898 \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```

4899 \ifdefined\eql@frame@cmd
4900 \eql@frame@measure
4901 \advance\eql@cellwidth@\eql@frame@margin@
4902 \advance\eql@prevwidth@\eql@frame@margin@
4903 \eql@frame@print
4904 \fi

```

update height and depth

```

4905 \ifdim\ht\eql@cellbox@>\eql@line@height@
4906 \global\eql@line@height@\ht\eql@cellbox@
4907 \fi
4908 \ifdim\dp\eql@cellbox@>\eql@line@depth@
4909 \global\eql@line@depth@\dp\eql@cellbox@
4910 \fi

```

print box

```

4911 \kern-\eql@prevwidth@
4912 \unhbox\eql@cellbox@
4913 \dimen@-\eql@cellwidth@
4914 \fi

```

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

```

4915 \advance\dimen@\eql@colwidth@get\eql@column@\relax
4916 \kern\dimen@

```

update available and used space

```

4917 \dimen@\eql@colwidth@get\eql@column@\relax
4918 \ifdim\eql@cellwidth@>z@
4919 \ifdim\eql@line@width@=z@
4920 \eql@line@avail@\eql@line@pos@
4921 \ifodd\eql@column@
4922 \advance\eql@line@avail@\dimen@
4923 \advance\eql@line@avail@-\eql@cellwidth@
4924 \fi
4925 \global\eql@line@avail@\eql@line@avail@
4926 \fi
4927 \eql@line@width@\eql@line@pos@
4928 \ifodd\eql@column@
4929 \advance\eql@line@width@\dimen@

```



```

4930     \else
4931         \advance\eql@line@width@\eql@cellwidth@
4932     \fi
4933     \global\eql@line@width@\eql@line@width@
4934 \fi
4935 \advance\eql@line@pos@\dimen@
4936 \ifodd\eql@column@\else
4937     \advance\eql@line@pos@\eql@colsep@
4938 \fi
4939 \global\eql@line@pos@\eql@line@pos@
4940 }

4941 \def\eql@columns@print@trailright{%
4942     &\omit
4943     \eql@prevwidth@\wd\eql@cellbox@
4944     \let\eql@frame@cmd\eql@frame@prevcmd
4945     \global\advance\eql@column@\@ne
4946     \eql@columns@print@cell
4947 }

```

lums@print@line@end

```

4948 \def\eql@columns@print@line@end{%
4949     \eql@punct@apply@line
4950     \eql@hook@lineout
4951 % \TODO add an even column with empty stuff if box processing deferred
4952     \ifodd\eql@column@
4953         \expandafter\eql@columns@print@trailright
4954     \fi
4955     \eql@columns@completerow
4956     \eql@columns@print@tag
4957 }

```

ql@columns@print@tag

```

4958 \def\eql@columns@print@tag{%
4959     \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4960     \eql@display@firstavail@set\eql@line@avail@
4961     \eql@columns@overfull
4962     \eql@numbering@print@line@eval
4963     \if@eqnsw
4964         \eql@tagbox@make\eql@composetag@print
4965     \fi
4966     \eql@tagpos@print@line@eval
4967     \eql@tagbox@print@cell
4968     \eql@tagpos@print@line@end
4969 }

```

\eql@columns@print

```

4970 \def\eql@columns@print{%
4971     \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
4972     \eql@shape@align@disable
4973     \eql@display@halign@init\eql@columns@print@line@begin
4974     \eql@multi@cr@let\eql@columns@print@line@end
4975     \ifdefined\eql@amp@mode
4976         \let&\eql@columns@amp

```

```

4977 \fi
4978 \tabskip\eq@marginleft@

4979 \halign{%
4980   &%
4981   \global\advance\eq@column@\@ne
4982   \global\let\eq@cell@container\@empty
4983   \global\setbox\eq@cellbox@\hbox{%
4984     \eq@strut@cell
4985     \@lign
4986     $\m@th\eq@mathstyle
4987     \eq@hook@colin
4988     ##%
4989     \eq@punct@apply@next
4990     \eq@class@innerleft
4991     \eq@hook@innerleft
4992     \eq@tagging@mathsave
4993     $%
4994     \eq@tagging@mathaddlast
4995   }%
4996   \eq@cell@container
4997   \hfil
4998   \eq@columns@print@cell
4999   \global\let\eq@frame@prevcmd\eq@frame@cmd
5000   \tabskip\z@skip
5001   &%
5002   \eq@prevwidth@\wd\eq@cellbox@
5003   \let\eq@frame@cmd\eq@frame@prevcmd
5004   \global\advance\eq@column@\@ne
5005   \global\let\eq@cell@container\@empty
5006   \setbox\eq@cellbox@\hbox{%
5007     \unhbox\eq@cellbox@
5008     \eq@strut@cell
5009     \@lign
5010     $\m@th\eq@mathstyle
5011     \eq@hook@innerright
5012     \eq@class@innerright@sel
5013     ##%
5014     \eq@punct@apply@col
5015     \eq@hook@colout
5016     \eq@tagging@mathsave
5017     $%
5018     \eq@tagging@mathaddlast
5019   }%
5020   \eq@cell@container
5021   \eq@columns@print@cell
5022   \hfil
5023   \tabskip\eq@colsep@\relax
5024 \crrr

5025 \noalign{%
5026   \eq@display@halign@start
5027   \eq@numbering@print@block@begin
5028   \eq@hook@blockbefore
5029 }%
5030 \eq@hook@blockin
5031 \eq@scan@body
5032 \ifvmode\else
5033   \relax

```

```

5034     \eql@punct@apply@block
5035     \eql@hook@blockout
5036     \eql@multi@endline
5037     \cr
5038   \fi
5039   \noalign{%
5040     \eql@hook@blockafter
5041     \eql@display@halign@end
5042     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
5043   }%
5044   \eql@tagging@tables@savealign
5045 }%
5046 }

```

15 Interface

15.1 Options Processing

TODO: describe

```

5047 \def\eql@equations@test@setopt{\let\eql@equations@test\eql@equations@testopt}
5048 \def\eql@equations@test@setall{\let\eql@equations@test\eql@equations@testall}

```

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

```

5049 \def\eql@equations@testopt#1{%
5050   \eql@testopt@tight{\eql@equations@testopt@set{#1}}{}}
5051 \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

```

5052 \def\eql@equations@testall{\eql@parseopt@env\eql@equations@testall@parse}
5053 \def\eql@equations@testall@parse{%
5054   \ifx\eql@parseopt@token*%
5055     \let\eql@parseopt@next\eql@parseopt@nonumber
5056   \fi
5057   \ifx\eql@parseopt@token!%
5058     \let\eql@parseopt@next\eql@parseopt@donumber
5059   \fi
5060   \ifx\eql@parseopt@token/%
5061     \let\eql@parseopt@next\eql@parseopt@transpose
5062   \fi
5063   \ifx\eql@parseopt@token[%
5064     \let\eql@parseopt@next\eql@parseopt@opt
5065   \fi
5066   \ifx\eql@parseopt@token\eql@atxi
5067     \let\eql@parseopt@next\eql@parseopt@label
5068   \fi
5069   \ifx\eql@parseopt@token\eql@atxii
5070     \let\eql@parseopt@next\eql@parseopt@label
5071   \fi
5072   \ifx\eql@parseopt@token.%
5073     \let\eql@parseopt@next\eql@parseopt@punctpass
5074   \fi

```

```

5075 \ifx\eql@parseopt@token,%
5076   \let\eql@parseopt@next\eql@parseopt@punctpass
5077 \fi
5078 \ifx\eql@parseopt@token~%
5079   \let\eql@parseopt@next\eql@parseopt@punctpass
5080 \fi
5081 \ifx\eql@parseopt@token'%
5082   \let\eql@parseopt@next\eql@parseopt@punctopt
5083 \fi
5084 \ifx\eql@parseopt@token-%
5085   \let\eql@parseopt@next\eql@parseopt@single
5086 \fi
5087 \ifx\eql@parseopt@token=%
5088   \let\eql@parseopt@next\eql@parseopt@lines
5089 \fi
5090 \ifx\eql@parseopt@token|%
5091   \let\eql@parseopt@next\eql@parseopt@columns
5092 \fi
5093 \ifx\eql@parseopt@token<%
5094   \let\eql@parseopt@next\eql@parseopt@ampeq
5095 \fi
5096 \ifx\eql@parseopt@token>%
5097   \let\eql@parseopt@next\eql@parseopt@eqamp
5098 \fi
5099 \ifx\eql@parseopt@token\label
5100   \let\eql@parseopt@next\eql@parseopt@end
5101 \fi
5102 \ifx\eql@parseopt@token\begin
5103   \let\eql@parseopt@next\eql@parseopt@end
5104 \fi
5105 }

```

equations@end@testall **TODO:** describe

```

5106 \def\eql@equations@end@testall{%
5107   \eql@parseopt@env\eql@equations@end@testall@parse}
5108 \def\eql@equations@end@testall@parse{%
5109   \ifx\eql@parseopt@token.%
5110     \let\eql@parseopt@next\eql@parseopt@punctpass
5111   \fi
5112   \ifx\eql@parseopt@token,%
5113     \let\eql@parseopt@next\eql@parseopt@punctpass
5114   \fi
5115   \ifx\eql@parseopt@token~%
5116     \let\eql@parseopt@next\eql@parseopt@punctpass
5117   \fi
5118   \ifx\eql@parseopt@token'%
5119     \let\eql@parseopt@next\eql@parseopt@punctblock
5120   \fi
5121 }

```

equations@processopt The macro `\eql@equations@processopt` processes the options received by `\eqnaddopt`. First, clear several non-persistent registers (labels, tags, direct vertical spacing). Then process the arguments. Finally evaluate `\eql@indent@val` and `\eql@tagsepmin@val` and prevent main punctuation from being passed to nested environments:

```

5122 \def\eql@equations@processopt{%
5123   \let\eql@tags@container@block\eql@tags@container@clear
5124   \let\eql@tags@frame@cmd\@firstofone

```

```

5125 \let\eql@skip@force@above\@undefined
5126 \let\eql@skip@force@below\@undefined
5127 \let\eql@skip@force@leave\@undefined
5128 \let\eql@display@linewidth\@undefined
5129 \let\eql@display@marginleft\@undefined
5130 \let\eql@display@marginright\@undefined
5131 \eql@abovespace@\z@skip
5132 \eql@belowspace@\z@skip
5133 \eql@displaybreak@prepen@\@MM
5134 \eql@displaybreak@postpen@\@MM
5135 \eql@nextopt@process{equations}%
5136 \let\eql@punct@next\@undefined
5137 \eql@indent@\glueexpr\eql@indent@val\relax
5138 \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
5139 }

```

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:

```

5140 \def\eql@single@cr@error{%
5141   \eql@error{Cannot use '\string\\' within display equation.
5142     Please switch to equations environment}}%
5143 }

```

`\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:

```

5144 \def\eql@single@start{%
5145   \eql@display@enter
5146   \eql@tagging@start
5147   \eql@dollar@dollar@begin
5148   \eql@display@adjust
5149   \eql@numbering@init
5150   \eql@stack@save@equations
5151   \eql@numbering@single@init
5152   \ifdefined\eql@single@cr@mode
5153     \let\\\eql@single@cr@mode
5154   \fi
5155   \ifdefined\eql@amp@mode
5156     \let&\eql@break@amp
5157   \fi
5158   \ifdefined\eql@single@native
5159     \let\eql@single@start@sel\eql@single@start@native
5160     \let\eql@single@end@sel\eql@single@end@native
5161   \else
5162     \let\eql@single@start@sel\eql@single@start@print
5163     \let\eql@single@end@sel\eql@single@end@print
5164   \fi
5165   \eql@single@start@sel
5166 }

```

`\eql@single@end` Closing code for single-line equation. Apply punctuation for the block, perform

mode-specific ending, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```

5167 \def\eql@single@end{%
5168   \eql@punct@apply@block
5169   \eql@hook@eqout
5170   \eql@single@end@sel
5171   \global\eql@punct@top@reset
5172   \eql@stack@restore
5173   \eql@dollar@dollar@end
5174   \eql@tagging@end
5175   \eql@display@leave
5176 }

```

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

```

5177 \def\eql@single@main{%
5178   \eql@single@start
5179   \eql@scan@body
5180   \eql@single@end
5181 }

```

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

```

5182 \def\eql@equations@single@set{%
5183   \ifdefined\eql@single@doscan
5184     \let\eql@equations@main\eql@single@main
5185   \else
5186     \let\eql@equations@main\@undefined
5187   \fi
5188 }

```

15.3 Multi-Line Main

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

```

5189 \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

```

5190 \def\eql@multi@main{%
5191   \eql@display@enter
5192   \eql@tagging@start
5193   \eql@dollar@dollar@begin
5194   \eql@display@adjust
5195   \eql@numbering@init
5196   \eql@stack@save@equations
5197   \ifdefined\eql@transpose@active
5198     \ifdefined\eql@multi@linesmode\else
5199       \eql@transpose
5200     \fi
5201   \fi
5202   \ifdefined\eql@numbering@subeq@use
5203     \eql@numbering@subeq@init
5204   \fi

```

```

5205 \eql@display@init
5206 \let\intertext\eql@intertext
5207 \let\endintertext\endeql@intertext
5208 \eql@shape@align@enable

```

Now measure the given multi-line equations body:

```

5209 \ifdefined\eql@multi@linesmode
5210   \eql@lines@measure
5211 \else
5212   \ifdefined\eql@ampproof@active
5213     \eql@ampproof
5214   \fi
5215   \eql@columns@measure
5216 \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`:

```

5217 \ifdefined\eql@numbering@subeq@use
5218   \eql@numbering@subeq@test
5219 \fi
5220 \ifdefined\eql@multi@linesmode\else
5221   \ifdefined\eql@multi@linesfallback
5222     \ifnum\eql@totalcolumns@=\@ne
5223       \let\eql@multi@linesmode\eql@true
5224       \ifx\eql@multi@linesfallback\z@\else
5225         \eql@lines@measure
5226       \fi
5227     \fi
5228   \fi
5229 \fi

```

Adjust the multi-line equations body:

```

5230 \ifdefined\eql@multi@linesmode
5231   \eql@lines@adjust
5232 \else
5233   \eql@columns@adjust
5234 \fi

```

Now print the multi-line equations body:

```

5235 \eql@display@print
5236 \eql@numbering@print@init
5237 \ifdefined\eql@multi@linesmode
5238   \eql@lines@print
5239 \else
5240   \eql@columns@print
5241 \fi
5242 \eql@display@close

```

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

```

5243 \ifdefined\eql@numbering@subeq@use
5244   \eql@numbering@subeq@close
5245 \fi
5246 \global\eql@punct@top@reset

```

```

5247 \eq\stack@restore
5248 \eq\dollardollar@end
5249 \eq>tagging@end
5250 \eq@display@leave
5251 }

```

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

`@equations@lines@set`

```

5252 \def\eq\equations@columns@set{%
5253   \let\eq\equations@main\eq@multi@main
5254   \let\eq@multi@linesmode\eq@false
5255 }
5256 \def\eq\equations@lines@set{%
5257   \let\eq\equations@main\eq@multi@main
5258   \let\eq@multi@linesmode\eq@true
5259 }

```

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 `\eq\equations@start`:

```

5260 \newenvironment{equations}{%
5261   \ifmmode
5262     \expandafter\eq\equations@env@cancel
5263   \else
5264     \eq@verbose@info\eq@verbose@msg@enterenv
5265     \expandafter\eq\equations@env@open
5266   \fi
5267 }{%
5268   \ifdefined\eq\equations@main\else
5269     \expandafter\eq@single@end
5270   \fi
5271   \ignorespacesafterend
5272   \eq@verbose@info\eq@verbose@msg@leaveenv
5273 }
5274 \eq@markline@amsthm@register{equations}
5275 \eq>tagging@register@luamml{equations}

```

`equations@env@cancel`

```

5276 \def\eq\equations@env@cancel{%
5277   \eq@error@mathmode{\string\begin{\@currenvir}}}%
5278   \let\eq@scan@call\eq@scan@env@cancel
5279   \eq@scan@env
5280 }

```

`equations@env@open`

```

5281 \def\eq\equations@env@open{%
5282   \ifdefined\eq\equations@env@modifier
5283     \eq\equations@test@setall

```



```

5284 \else
5285     \eql@equations@nomodifier
5286 \fi
5287 \eql@ampprotect\eql@equations@test\eql@equations@env@start
5288 }

```

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

```

5289 \def\eql@equations@env@start{%
5290     \eql@equations@processopt
5291     \ifdefined\eql@equations@main
5292         \eql@equations@call@set
5293         \expandafter\eql@scan@env
5294     \else
5295         \expandafter\eql@single@start
5296     \fi
5297 }

```

`\eql@equations@call`

```

5298 \def\eql@equations@call{\eql@equations@main\eql@scan@end}
5299 \def\eql@equations@call@test{%
5300     \eql@ampprotect\eql@equations@end@testall\eql@equations@call}
5301 \def\eql@equations@call@set{%
5302     \ifdefined\eql@equations@end@modifier
5303         \let\eql@scan@call\eql@equations@call@test
5304     \else
5305         \let\eql@scan@call\eql@equations@call
5306     \fi
5307 }

```

Square Brackets.

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

```

5308 \newenvironment{equations@sqr}{}{}
5309 \eql@markline@amsthm@register{equations@sqr}
5310 \eql@tagging@register@luamml{equations@sqr}

```

`\eql@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`:

```

5311 \def\eql@equations@sqr@open{%
5312     \expandafter\eqnadopt\expandafter{\eql@equations@sqr@opt}%
5313     \begin{equations@sqr}%
5314     \eql@verbose@info\eql@verbose@msg@enterenv
5315     \let\]\eql@equations@sqr@close
5316     \ifdefined\eql@equations@sqr@modifier
5317         \eql@equations@test@setall
5318     \else
5319         \eql@equations@nomodifier
5320     \fi
5321     \eql@ampprotect\eql@equations@test\eql@equations@sqr@start
5322 }

```

`@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`:

```
5323 \def\eql@equations@sqr@start{%
5324   \eql@equations@processopt
5325   \ifdefined\eql@equations@main
5326     \eql@equations@call@set
5327     \expandafter\eql@scan@sqr
5328   \else
5329     \expandafter\eql@single@start
5330   \fi
5331 }
```

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

```
5332 \protected\def\eql@equations@sqr@close{%
5333   \ifdefined\eql@equations@main
5334     \let\eql@equations@end@modifier\eql@false
5335   \fi
5336   \ifdefined\eql@equations@end@modifier
5337     \expandafter\eql@ampprotect\expandafter\eql@equations@end@testall
5338   \fi
5339   \eql@equations@sqr@end
5340 }
```

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

```
5341 \def\eql@equations@sqr@end{%
5342   \ifdefined\eql@equations@main\else
5343     \expandafter\eql@single@end
5344   \fi
5345   \eql@verbose@info\eql@verbose@msg@leaveenv
5346   \end{equations@sqr}%
5347   \ignorespaces
5348 }
```

Angle Brackets.

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

```
5349 \newenvironment{equations@ang}{}{}
5350 \eql@markline@amsthm@register{equations@ang}
5351 \eql@tagging@register@luamml{equations@ang}
```

`\eql@equations@ang@open`

```
5352 \def\eql@equations@ang@open{%
5353   \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
5354   \begin{equations@ang}%
5355   \eql@verbose@info\eql@verbose@msg@enterenv
5356   \let\>\eql@equations@ang@close
5357   \ifdefined\eql@equations@ang@modifier
5358     \eql@equations@test@setall
5359   \else
5360     \eql@equations@nomodifier
5361   \fi
5362   \eql@ampprotect\eql@equations@test\eql@equations@ang@start
5363 }
```

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

```
5364 \def\eq@equations@ang@start{%
5365   \eq@equations@processopt
5366   \ifdefined\eq@equations@main
5367     \eq@equations@call@set
5368     \expandafter\eq@scan@ang
5369   \else
5370     \expandafter\eq@single@start
5371   \fi
5372 }
```

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

```
5373 \def\eq@equations@ang@close{%
5374   \ifdefined\eq@equations@main
5375     \let\eq@equations@end@modifier\eq@false
5376   \fi
5377   \ifdefined\eq@equations@end@modifier
5378     \expandafter\eq@ampprotect\expandafter\eq@equations@end@testall
5379   \fi
5380   \eq@equations@ang@end
5381 }
```

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

```
5382 \def\eq@equations@ang@end{%
5383   \ifdefined\eq@equations@main\else
5384     \expandafter\eq@single@end
5385   \fi
5386   \eq@verbose@info\eq@verbose@msg@leaveenv
5387   \end{equations@ang}%
5388   \ignorespaces
5389 }
```

16 Options

16.1 Selection Tools

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

```
5390 \def\eq@decide@abovebelow#1#2#3#4#5{%
5391   \eq@decide@select{#1}{#2}{#3}{%
5392     {abovebelow,both,tb}{#4#5},%
5393     {above,top,t}{#4},%
5394     {below,bottom,b}{#5}}}
```

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

```
5395 \def\eq@decide@situation#1#2#3#4{%
5396   \eq@decide@select{#1}{#2}{#3}{%
5397     {{long}}{\def#4{0}}},%
5398     {{short}}{\def#4{1}}},%
5399     {{cont}}{\def#4{2}}},%
5400     {{par}}{\def#4{3}}},%
5401     {{top}}{\def#4{4}}},%
```

```

5402    {\noskip}{\def#4{5}}},%
5403    {\medskip}{\def#4{6}}}}

```

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

```

5404 \def\eqL@decide@delim#1#2#3{%
5405   \eqL@decide@select{#1}{#2}{#3}{%
5406     {\,,\eqL@decide@false}{\eqL@box@wrap{}{}},%
5407     {\eqL@decide@true,r,round}{\eqL@box@delim()},%
5408     {\s,sqr,square}{\eqL@box@delim[]{}},%
5409     {\c,curly,braces}{\eqL@box@delim\lbrace\rbrace},%
5410     {\a,ang,angle}{\eqL@box@delim\langle\rangle},%
5411     {\v,vert}{\eqL@box@delim\vert\vert},%
5412     {\d,dvert}{\eqL@box@delim\Vert\Vert},%
5413     {\relax{\eqL@box@delim#3}}}%

```

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:

```

5414 \eqL@define@key{box}{gathered,gather,ga,lines,ln}[]{%
5415   \eqL@box@lines@set}
5416 \eqL@define@key{box}{aligned,align,al,columns,col}[]{%
5417   \eqL@box@columns@set}
5418 \eqL@define@key{box}{cases}[]{%
5419   \eqL@box@cases@set\eqL@box@ldelim\lbrace%
5420   \def\eqL@box@colsep{\eqL@box@condsep}}
5421 \eqL@define@key{box}{matrix}[r]{%
5422   \eqL@box@lines@set\eqL@shape@set{center}%
5423   \let\eqL@spread@reset\eqL@true\def\eqL@spread@val{\z@}%
5424   \def\eqL@box@colsep{\eqL@box@shortsep}%
5425   \let\eqL@mathstyle\@empty
5426   \eqL@punct@clear
5427   \eqL@box@cr@test@setopt
5428   \eqL@box@amp@test@setopt
5429   \let\eqL@box@end@modifier\eqL@false
5430   \eqL@decide@delim{#3}{#2}{#1}}
5431 \eqL@define@key{box}{top,t}[]{\let\eqL@box@box\vtop}
5432 \eqL@define@key{box}{center,c}[]{\let\eqL@box@box\vcenter}
5433 \eqL@define@key{box}{bottom,b}[]{\let\eqL@box@box\vbbox}
5434 \eqL@define@key{box}{intro}{%
5435   \def\eqL@box@cases@condintro{#1}}
5436 \eqL@define@key{box}{introtext}{%
5437   \def\eqL@box@cases@condintro{%
5438     \ifmmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}
5439 \eqL@define@key{box}{textcond}[true]{%
5440   \eqL@decide@select{#3}{#2}{#1}{%
5441     {\eqL@decide@true,text}{\let\eqL@box@cases@condtext\eqL@true}},%
5442     {\eqL@decide@false,math}{\let\eqL@box@cases@condtext\eqL@false}}}%
5443 \eqL@define@key{setup}{scanbox}[true]{%
5444   \eqL@decide@bool{#3}{#2}{#1}\eqL@box@doscan}
5445 \eqL@define@key{box}{scan}[true]{%
5446   \eqL@decide@bool{#3}{#2}{#1}\eqL@box@doscan}
5447 \eqL@define@key{setup}{boxangopt}[]{%

```

```
5448 \def\eql@box@ang@opt{columns,#1}}
```

Modes for Equations Environment.

```
5449 \let\eql@box@doscans\eqn@false
```

Declare modes and switches for the equations environment:

```
5450 \eql@define@key{equations}{equation,eq,single,1}[]{\eqlequations@single@set}
5451 \eql@define@key{equations}{gathered,gather,ga,lines,ln}[]{%
5452   \eqlequations@lines@set}
5453 \eql@define@key{equations}{aligned,align,al,columns,col}[]{%
5454   \eqlequations@columns@set}
5455 \eql@define@key{equations,setup}{transpose}[true]{%
5456   \eql@decide@select{#3}{#2}{#1}{%
5457     {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
5458     {\noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
5459     {\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}%
5460 \eql@define@key{setup}{nativeequation}[true]{%
5461   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
5462 \eql@define@key{equations}{native}[true]{%
5463   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
5464   \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
5465 \eql@define@key{setup}{scanequation}[true]{%
5466   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscans}
5467 \eql@define@key{equations}{scan}[true]{%
5468   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscans}
5469 \eql@define@key{setup}{sqropt}[]{%
5470   \def\eqlequations@sqr@opt{equation,#1}}
5471 \eql@define@key{setup}{angopt}[]{%
5472   \def\eqlequations@ang@opt{columns,#1}}
```

TODO: describe

```
5473 \eql@define@key{control}{restoreexterior}[]{\eql@display@restore}
5474 \eql@define@key{control}{restoreexterior*}[]{\@arrayparboxrestore}
```

Modes for Modifier Scanning.

```
5475 \def\eqlequations@nomodifier{%
5476   \eqlequations@test@setopt
5477   \let\eqlequations@end@modifier\eql@false
5478   \eql@multi@cr@test@setopt
5479   \eql@columns@amp@test@setopt
5480   \eql@break@cr@test@setopt
5481   \eql@break@amp@test@setopt
5482 }
5483 \let\eqlequations@env@modifier\eql@false
5484 \let\eqlequations@sqr@modifier\eql@true
5485 \let\eqlequations@ang@modifier\eql@true
5486 \let\eqlequations@end@modifier\eql@true
5487 \eqlequations@test@setall
5488 \eql@multi@cr@test@setall
5489 \eql@columns@amp@test@setall
5490 \eql@break@cr@test@setall
5491 \eql@break@amp@test@setall
5492 \eql@break@test@setopt
5493 \let\eql@multi@cr@relnext\eql@true
```

```

5494 \def\eq@box@nomodifier{%
5495   \eq@box@test@setopt
5496   \let\eq@box@end@modifier\eq@false
5497   \eq@box@cr@test@setopt
5498   \eq@box@amp@test@setopt
5499 }
5500 \let\eq@box@env@modifier\eq@false
5501 \let\eq@box@ang@modifier\eq@true
5502 \let\eq@box@end@modifier\eq@true
5503 \eq@box@test@setall
5504 \eq@box@cr@test@setall
5505 \eq@box@amp@test@setall

5506 \eq@define@key{setup}{modifier}[true]{%
5507   \eq@decide@if{#3}{#2}{#1}%
5508   {\let\eq@equations@env@modifier\eq@true
5509     \let\eq@equations@sqr@modifier\eq@true
5510     \let\eq@equations@ang@modifier\eq@true
5511     \let\eq@box@env@modifier\eq@true
5512     \let\eq@box@ang@modifier\eq@true
5513     \eq@multi@cr@test@setall\eq@break@cr@test@setall
5514     \eq@columns@amp@test@setall\eq@break@amp@test@setall
5515     \eq@box@cr@test@setall\eq@box@amp@test@setall
5516     \eq@break@test@setall}%
5517   {\let\eq@equations@env@modifier\eq@false
5518     \let\eq@equations@sqr@modifier\eq@false
5519     \let\eq@equations@ang@modifier\eq@false
5520     \let\eq@box@env@modifier\eq@false
5521     \let\eq@box@ang@modifier\eq@false
5522     \eq@multi@cr@test@setopt\eq@break@cr@test@setopt
5523     \eq@columns@amp@test@setopt\eq@break@amp@test@setopt
5524     \eq@box@cr@test@setopt\eq@box@amp@test@setopt
5525     \eq@break@test@setopt}}
5526 \eq@define@key{setup}{modifierenv}[true]{%
5527   \eq@decide@bool{#3}{#2}{#1}\eq@equations@env@modifier
5528   \eq@decide@bool{#3}{#2}{#1}\eq@box@env@modifier}
5529 \eq@define@key{setup}{modifiersqrang}[true]{%
5530   \eq@decide@bool{#3}{#2}{#1}\eq@equations@sqr@modifier
5531   \eq@decide@bool{#3}{#2}{#1}\eq@equations@ang@modifier
5532   \eq@decide@bool{#3}{#2}{#1}\eq@box@ang@modifier}
5533 \eq@define@key{setup}{modifierend}[true]{%
5534   \eq@decide@bool{#3}{#2}{#1}\eq@equations@end@modifier
5535   \eq@decide@bool{#3}{#2}{#1}\eq@box@end@modifier}
5536 \eq@define@key{setup}{modifierreqncr}[true]{\eq@decide@if{#3}{#2}{#1}%
5537   {\eq@multi@cr@test@setall\eq@break@cr@test@setall
5538     \eq@box@cr@test@setall}%
5539   {\eq@multi@cr@test@setopt\eq@break@cr@test@setopt
5540     \eq@box@cr@test@setopt}}
5541 \eq@define@key{setup}{modifierreqnamp}[true]{\eq@decide@if{#3}{#2}{#1}%
5542   {\eq@columns@amp@test@setall\eq@break@amp@test@setall
5543     \eq@box@amp@test@setall}%
5544   {\eq@columns@amp@test@setopt\eq@break@amp@test@setopt
5545     \eq@box@amp@test@setopt}}
5546 \eq@define@key{setup}{modifierbreak}[true]{\eq@decide@if{#3}{#2}{#1}%
5547   \eq@break@test@setall\eq@break@test@setopt}
5548 \eq@define@key{setup}{modifierwarning}[all]{%
5549   \eq@decide@select{#3}{#2}{#1}{%
5550     {\eq@decide@false{\let\eq@parseopt@warn@env\@empty
5551       \let\eq@parseopt@warn@cr\@empty}},%

```

```

5552   {{env,-}}{\let\eql@parseopt@warn@env\eql@warn@parseopt
5553   \let\eql@parseopt@warn@cr@empty}},%
5554   {{all,\eql@decide@true}}{\let\eql@parseopt@warn@env\eql@warn@parseopt
5555   \let\eql@parseopt@warn@cr\eql@warn@parseopt}},%
5556   {{verbose,+}}{\let\eql@parseopt@warn@env\eql@warn@parseopt@verbose
5557   \let\eql@parseopt@warn@cr\eql@warn@parseopt@verbose}}}}

5558 \eql@define@key{setup}{modifier@eqn}[true]{%
5559   \eql@decide@bool{#3}{#2}{#1}\eql@equations@env@modifier
5560   \eql@decide@bool{#3}{#2}{#1}\eql@equations@sqr@modifier
5561   \eql@decide@bool{#3}{#2}{#1}\eql@equations@ang@modifier}
5562 \eql@define@key{setup}{modifier@eqnenv}[true]{%
5563   \eql@decide@bool{#3}{#2}{#1}\eql@equations@env@modifier}
5564 \eql@define@key{setup}{modifier@eqnsqr}[true]{%
5565   \eql@decide@bool{#3}{#2}{#1}\eql@equations@sqr@modifier}
5566 \eql@define@key{setup}{modifier@eqnang}[true]{%
5567   \eql@decide@bool{#3}{#2}{#1}\eql@equations@ang@modifier}
5568 \eql@define@key{setup}{modifier@eqnend}[true]{%
5569   \eql@decide@bool{#3}{#2}{#1}\eql@equations@end@modifier}
5570 \eql@define@key{setup}{modifier@eqncr}[true]{\eql@decide@if{#3}{#2}{#1}%
5571   {\eql@multi@cr@test@setall\eql@break@cr@test@setall}}%
5572   {\eql@multi@cr@test@setopt\eql@break@cr@test@setopt}}
5573 \eql@define@key{setup}{modifier@eqnamp}[true]{\eql@decide@if{#3}{#2}{#1}%
5574   {\eql@columns@amp@test@setall\eql@break@amp@test@setall}}%
5575   {\eql@columns@amp@test@setopt\eql@break@amp@test@setopt}}

5576 \eql@define@key{setup}{modifier@box}[true]{%
5577   \eql@decide@bool{#3}{#2}{#1}\eql@box@env@modifier
5578   \eql@decide@bool{#3}{#2}{#1}\eql@box@ang@modifier}
5579 \eql@define@key{setup}{modifier@boxenv}[true]{%
5580   \eql@decide@bool{#3}{#2}{#1}\eql@box@env@modifier}
5581 \eql@define@key{setup}{modifier@boxang}[true]{%
5582   \eql@decide@bool{#3}{#2}{#1}\eql@box@ang@modifier}
5583 \eql@define@key{setup}{modifier@boxend}[true]{%
5584   \eql@decide@bool{#3}{#2}{#1}\eql@box@end@modifier}
5585 \eql@define@key{setup}{modifier@boxcr}[true]{%
5586   \eql@decide@if{#3}{#2}{#1}\eql@box@cr@test@setall\eql@box@cr@test@setopt}
5587 \eql@define@key{setup}{modifier@boxamp}[true]{%
5588   \eql@decide@if{#3}{#2}{#1}\eql@box@amp@test@setall\eql@box@amp@test@setopt}}

5589 \eql@define@key{equations}{modifier@end}[true]{%
5590   \eql@decide@bool{#3}{#2}{#1}\eql@equations@end@modifier}
5591 \eql@define@key{equations}{modifier@cr}[true]{\eql@decide@if{#3}{#2}{#1}%
5592   {\eql@multi@cr@test@setall\eql@break@cr@test@setall}}%
5593   {\eql@multi@cr@test@setopt\eql@break@cr@test@setopt}}
5594 \eql@define@key{equations}{modifier@amp}[true]{\eql@decide@if{#3}{#2}{#1}%
5595   {\eql@columns@amp@test@setall\eql@break@amp@test@setall}}%
5596   {\eql@columns@amp@test@setopt\eql@break@amp@test@setopt}}
5597 \eql@define@key{setup,equations}{cr@relnext}[true]{%
5598   \eql@decide@bool{#3}{#2}{#1}\eql@multi@cr@relnext}

5599 \eql@define@key{box}{modifier}[true]{\eql@decide@if{#3}{#2}{#1}%
5600   {\eql@box@cr@test@setall\eql@box@amp@test@setall
5601   \let\eql@box@end@modifier\eql@true}}%
5602   {\eql@box@cr@test@setopt\eql@box@amp@test@setopt
5603   \let\eql@box@end@modifier\eql@false}}
5604 \eql@define@key{box}{modifier@end}[true]{%
5605   \eql@decide@bool{#3}{#2}{#1}\eql@box@end@modifier}
5606 \eql@define@key{box}{modifier@cr}[true]{%

```

```

5607 \eqld@decide@if{#3}{#2}{#1}\eqld@box@cr@test@setall\eqld@box@cr@test@setopt}
5608 \eqld@define@key{box}{modifieramp}[true]{%
5609 \eqld@decide@if{#3}{#2}{#1}\eqld@box@amp@test@setall\eqld@box@amp@test@setopt}

```

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

```

5610 \def\eqld@keycat{equations,box,setup}
5611 \eqld@define@key\eqld@keycat{spread}{%
5612 \let\eqld@spread@reset\eqld@false\def\eqld@spread@val{#1}}
5613 \eqld@define@key\eqld@keycat{spread*}[Opt]{%
5614 \let\eqld@spread@reset\eqld@true\def\eqld@spread@val{#1}}
5615 \eqld@define@key\eqld@keycat{strut}[true]{\eqld@decide@select{#3}{#2}{#1}{%
5616 {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%
5617 {{cell}{\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\relax}},%
5618 {{tag}{\let\eqld@strut@cell\relax\let\eqld@strut@tag\eqld@strut}},%
5619 {\eqld@decide@true
5620 {\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\eqld@strut}}}}
5621 \eqld@define@key{setup}{strutdepth}{\def\eqld@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

5622 \eqld@define@key\eqld@keycat{displayheight}[strut]{%
5623 \eqld@decide@select{#3}{#2}{#1}{%
5624 {\eqld@decide@false{\let\eqld@display@height\@undefined}},%
5625 {{strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
5626 {\relax{\def\eqld@display@height{#1}}}}
5627 \eqld@define@key\eqld@keycat{displaydepth}[strut]{%
5628 \eqld@decide@select{#3}{#2}{#1}{%
5629 {\eqld@decide@false{\let\eqld@display@depth\@undefined}},%
5630 {{strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
5631 {\relax{\def\eqld@display@depth{#1}}}}

```

Settings concerning page breaks:

```

5632 \eqld@define@key{equations}{prebreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
5633 {{force,4,\eqld@decide@true}{\eqld@displaybreak@pre4}},%
5634 {{high,3}{\eqld@displaybreak@pre3}},%
5635 {{med,medium,2}{\eqld@displaybreak@pre2}},%
5636 {{low,1}{\eqld@displaybreak@pre1}},%
5637 {{0,\eqld@decide@false}{\eqld@displaybreak@pre0}},%
5638 {{default,inherit,-1}{\eqld@displaybreak@pre\m@ne}}}}
5639 \eqld@define@key{equations}{postbreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
5640 {{force,4,\eqld@decide@true}{\eqld@displaybreak@post4}},%
5641 {{high,3}{\eqld@displaybreak@post3}},%
5642 {{med,medium,2}{\eqld@displaybreak@post2}},%
5643 {{low,1}{\eqld@displaybreak@post1}},%
5644 {{0,\eqld@decide@false}{\eqld@displaybreak@post0}},%
5645 {{default,inherit,-1}{\eqld@displaybreak@post\m@ne}}}}
5646 \eqld@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
5647 \eqld@decide@select{#3}{#2}{#1}{%
5648 {{full,4}{\eqld@displaybreak@inter4}},%
5649 {{high,3}{\eqld@displaybreak@inter3}},%
5650 {{med,medium,2}{\eqld@displaybreak@inter2}},%
5651 {{low,1}{\eqld@displaybreak@inter1}},%
5652 {{0,\eqld@decide@false}{\eqld@displaybreak@inter\z@}}}}
5653 \eqld@define@key{equations}{prepenalty}{%
5654 \eqld@displaybreak@prepen\@numexpr#1\relax}
5655 \eqld@define@key{equations}{postpenalty}{%

```



```

5656 \eqldisplaybreak@postpen@numexpr#1\relax}
5657 \eqld@define@key{equations,setup}{interpenalty}{%
5658 \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

5659 \eqld@define@key{control}{vspace}[]{\eqlvspace@add{#1}}
5660 \eqld@define@key{control}{vspace*}[]{\eqlvspace@addfixedbefore{#1}}
5661 \eqld@define@key{control}{vspace!}[]{\eqlvspace@addfixedafter{#1}}
5662 \eqld@define@key{control}{break}[4]{\eqldisplaybreak@level[{#1}]}
5663 \eqld@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...

```

5664 \eqld@define@key{equations}{noskip}[both]{%
5665 \eqld@decide@abovebelow{#3}{#2}{#1}%
5666 {\def\eqld@skip@force@above{5}}%
5667 {\def\eqld@skip@force@below{5}}}
5668 \eqld@define@key{equations}{short}[above]{%
5669 \eqld@decide@abovebelow{#3}{#2}{#1}%
5670 {\def\eqld@skip@force@above{1}}%
5671 {\def\eqld@skip@force@below{1}}}
5672 \eqld@define@key{equations}{long}[both]{%
5673 \eqld@decide@abovebelow{#3}{#2}{#1}%
5674 {\def\eqld@skip@force@above{0}}%
5675 {\def\eqld@skip@force@below{0}}}
5676 \eqld@define@key{equations}{medskip}[both]{%
5677 \eqld@decide@abovebelow{#3}{#2}{#1}%
5678 {\def\eqld@skip@force@above{6}}%
5679 {\def\eqld@skip@force@below{6}}}
5680 \eqld@define@key{equations}{par}[par]{%
5681 \eqld@decide@select{#3}{#2}{#1}{%
5682 {{default,\eqld@decide@false}}{\let\eqld@skip@force@leave\undefined}},%
5683 {{cont,hmode}}{\let\eqld@skip@force@leave\z@}},%
5684 {{par,vmode}}{\let\eqld@skip@force@leave@one
5685 \ifdefined\eqld@skip@force@below\else
5686 \def\eqld@skip@force@below{3}%
5687 \fi}},%
5688 {{top}}{\let\eqld@skip@force@leave\tw@
5689 \ifdefined\eqld@skip@force@below\else
5690 \def\eqld@skip@force@below{4}
5691 \fi}}}}

```

Specify vertical spacing explicitly:

```

5692 \eqld@define@key{equations}{skip}{%
5693 \def\eqld@skip@force@above{7}%
5694 \def\eqld@skip@custom@above{#1}%
5695 \let\eqld@skip@force@below\eqld@skip@force@above
5696 \let\eqld@skip@custom@below\eqld@skip@custom@above}
5697 \eqld@define@key{equations}{aboveskip}{%
5698 \def\eqld@skip@force@above{7}%
5699 \def\eqld@skip@custom@above{#1}}
5700 \eqld@define@key{equations}{belowskip}{%
5701 \def\eqld@skip@force@below{7}%
5702 \def\eqld@skip@custom@below{#1}}
5703 \eqld@define@key{equations}{abovespace}{%
5704 \advance\eqld@abovespace@glueexpr#1\relax}
5705 \eqld@define@key{equations}{belowspace}{%

```

5706 \advance\eql@belowspace@\glueexpr#1\relax}

Vertical spacing for intertext:

```

5707 \eql@define@key{intertext}{skip}{%
5708   \def\eql@skip@force@above{7}%
5709   \def\eql@skip@custom@above{#1}%
5710   \let\eql@skip@force@below\eql@skip@force@above
5711   \let\eql@skip@custom@below\eql@skip@custom@above}
5712 \eql@define@key{intertext}{aboveskip}{%
5713   \def\eql@skip@force@below{7}%
5714   \def\eql@skip@custom@below{#1}}
5715 \eql@define@key{intertext}{belowskip}{%
5716   \def\eql@skip@force@above{7}%
5717   \def\eql@skip@custom@above{#1}}
5718 \eql@define@key{intertext}{noskip}[both]{%
5719   \eql@decide@abovebelow{#3}{#2}{#1}%
5720   {\def\eql@skip@force@below{5}}%
5721   {\def\eql@skip@force@above{5}}}
5722 \eql@define@key{intertext}{short}[both]{%
5723   \eql@decide@abovebelow{#3}{#2}{#1}%
5724   {\def\eql@skip@force@below{1}}%
5725   {\def\eql@skip@force@above{1}}}
5726 \eql@define@key{intertext}{long}[both]{%
5727   \eql@decide@abovebelow{#3}{#2}{#1}%
5728   {\def\eql@skip@force@below{0}}%
5729   {\def\eql@skip@force@above{0}}}
5730 \eql@define@key{intertext}{medskip}[both]{%
5731   \eql@decide@abovebelow{#3}{#2}{#1}%
5732   {\def\eql@skip@force@below{6}}%
5733   {\def\eql@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

5734 \eql@define@key{setup}{skip,longskip}{%
5735   \abovedisplayskip\glueexpr#1\relax
5736   \belowdisplayskip\abovedisplayskip
5737   \def\eql@skip@long@above{#1}%
5738   \let\eql@skip@long@below\eql@skip@long@above}
5739 \eql@define@key{setup}{aboveskip,abovelongskip}{%
5740   \abovedisplayskip\glueexpr#1\relax
5741   \def\eql@skip@long@above{#1}}
5742 \eql@define@key{setup}{belowskip,belowlongskip}{%
5743   \belowdisplayskip\glueexpr#1\relax
5744   \def\eql@skip@long@below{#1}}
5745 \eql@define@key{setup}{aboveshortskip}{%
5746   \abovedisplayshortskip\glueexpr#1\relax
5747   \def\eql@skip@short@above{#1}}
5748 \eql@define@key{setup}{belowshortskip}{%
5749   \belowdisplayshortskip\glueexpr#1\relax
5750   \def\eql@skip@short@below{#1}}
5751 \eql@define@key{setup}{tagskip}{%
5752   \def\eql@skip@tag@above{#1}%
5753   \let\eql@skip@tag@below\eql@skip@tag@above}
5754 \eql@define@key{setup}{abovetagskip}{%
5755   \def\eql@skip@tag@above{#1}}
5756 \eql@define@key{setup}{belowtagskip}{%
5757   \def\eql@skip@tag@below{#1}}
5758 \eql@define@key{setup}{medskip}{%
5759   \def\eql@skip@med@above{#1}%

```

```

5760 \let\eq@skip@med@below\eq@skip@med@above}
5761 \eq@define@key{setup}{abovemedskip}{%
5762   \def\eq@skip@med@above{#1}}
5763 \eq@define@key{setup}{belowmedskip}{%
5764   \def\eq@skip@med@below{#1}}
5765 \eq@define@key{setup}{abovetopskip}{%
5766   \def\eq@skip@top@above{#1}}
5767 \eq@define@key{setup}{belowtopskip}{%
5768   \def\eq@skip@top@below{#1}}
5769 \eq@define@key{setup}{aboveparskip}{%
5770   \def\eq@skip@par@above{#1}}
5771 \eq@define@key{setup}{belowparskip}{%
5772   \def\eq@skip@par@below{#1}}
5773 \eq@define@key{setup}{abovecontskip}{%
5774   \eq@decide@select{#3}{#2}{#1}{%
5775     {{hide}{\def\eq@skip@cont@above{\eq@spread@val-\eq@skip@long@below}}},%
5776     {\relax{\def\eq@skip@cont@above{#1}}}}}
5777 \eq@define@key{setup}{belowcontskip}{%
5778   \def\eq@skip@cont@below{#1}}
5779 \eq@define@key{setup}{shortmode}{%
5780   \eq@decide@select{#3}{#2}{#1}{%
5781     {{\eq@decide@false,never}{\def\eq@skip@mode@short{0}}},%
5782     {{above,neverbelow,belowoff}{\def\eq@skip@mode@short{1}}},%
5783     {{belowone,belowsingle}{\def\eq@skip@mode@short{2}}},%
5784     {{belowall,always,on}{\def\eq@skip@mode@short{3}}}}}
5785 \eq@define@key{setup}{abovecontmode}{%
5786   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
5787 \eq@define@key{setup}{belowcontmode}{%
5788   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
5789 \eq@define@key{setup}{aboveparmode}{%
5790   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@above}
5791 \eq@define@key{setup}{belowparmode}{%
5792   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@below}
5793 \eq@define@key{setup}{abovetopmode}{%
5794   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@above}
5795 \eq@define@key{setup}{belowtopmode}{%
5796   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@below}

```

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

```

5797 \def\eq@keycat{equations,subequations}
5798 \eq@define@key\eq@keycat{label}{\eq@tags@addblock@label{#1}}
5799 \eq@define@key\eq@keycat{labelname}{\eq@tags@addblock@name{#1}}
5800 \eq@define@key\eq@keycat{tag}{\eq@tags@addblock@tag{#1}}
5801 \eq@define@key\eq@keycat{tag*}{%
5802   \eq@tags@addblock@tagform@off\eq@tags@addblock@tag{#1}}
5803 \eq@define@key\eq@keycat{taglabel}{\eq@tags@addblock@ref{#1}}

```

TODO: describe

```

5804 \eq@define@key{control}{label}{\eq@tags@add@label{#1}}
5805 \eq@define@key{control}{labelname}{\eq@tags@add@name{#1}}
5806 \eq@define@key{control}{tag}{\eq@tags@add@tag{#1}}
5807 \eq@define@key{control}{tag*}{\eq@tags@add@tagform@off\eq@tags@add@tag{#1}}
5808 \eq@define@key{control}{taglabel}{\eq@tags@add@ref{#1}}
5809 \eq@define@key{control}{shifttag}{\eq@tags@add@raiseshift{#1}}
5810 \eq@define@key{control}{smashtag}{\eq@tags@add@raisesmash{#1}}
5811 \eq@define@key{control}{pushtag}{\eq@tags@add@forceraise}

```

TODO: describe

```

5812 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
5813 \eqld@define@key{setup}{autolabel}[true]{%
5814   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}
5815 \eqld@define@key{setup}{autotag}[true]{%
5816   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

5817 \def\eqld@keycat{equations,setup}
5818 \eqld@define@key\eqld@keycat{tagmargin}[auto]{%
5819   \eqld@decide@select{#3}{#2}{#1}{%
5820     {\auto,\eqld@decide@false}{\let\eqld@tagmargin@val\undefined}},%
5821     {\relax{\def\eqld@tagmargin@val{#1}}}}
5822 \eqld@define@key\eqld@keycat{tagmargin*}{%
5823   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
5824 \eqld@define@key\eqld@keycat{tagmarginratio}{%
5825   \eqld@tagmargin@ratio@dimexpr#1pt\relax}
5826 \eqld@define@key\eqld@keycat{tagmarginthreshold}{%
5827   \def\eqld@tagmargin@threshold{#1}}
5828 \eqld@define@key\eqld@keycat{mintagsep}{\def\eqld@tagsepmin@val{#1}}
5829 \eqld@define@key\eqld@keycat{mintagwidth}{%
5830   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
5831 \eqld@define@key\eqld@keycat{mintagwidth*}{\settowidth\eqld@tagwidthmin@{#1}}
5832 \eqld@define@key\eqld@keycat{tagsnap}{%
5833   \eqld@decide@select{#3}{#2}{#1}{%
5834     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
5835     {\relax{\def\eqld@tagpos@snap{#1}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

5836 \def\eqld@keycat{equations,setup}
5837 \eqld@define@key\eqld@keycat{tagbox,taglayout}{%
5838   \eqld@tags@taglayout@set{#1}}
5839 \eqld@define@key\eqld@keycat{tagbox*,taglayout*}{%
5840   \eqld@tags@taglayout@set@direct{#1}}
5841 \eqld@define@key\eqld@keycat{tagform}{%
5842   \eqld@tags@tagform@set{#1}}
5843 \eqld@define@key\eqld@keycat{tagform*}{%
5844   \eqld@tags@tagform@set@direct{#1}}
5845 \eqld@define@key\eqld@keycat{subeqtemplate}{%
5846   \def\eqld@subequations@template####1####2{#1}%
5847   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}

5848 \eqld@define@key{control}{tagbox,taglayout}{%
5849   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
5850 \eqld@define@key{control}{tagbox*,taglayout*}{%
5851   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
5852 \eqld@define@key{control}{tagform}{%
5853   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
5854 \eqld@define@key{control}{tagform*}{[####1]}{%
5855   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

5856 \def\eqld@keycat{equations,setup}
5857 \eqld@define@key\eqld@keycat{numberline,number,num,numline,n}[all]{%

```

```

5858 \eql@decide@select{#3}{#2}{#1}{%
5859   {{\eql@decide@false,0,*}{\let\eql@numbering@active\eql@false}},%
5860   {{\eql@decide@true,!}{\let\eql@numbering@active\eql@true}},%
5861   {{none,n,-}{\let\eql@numbering@mode\eql@numbering@mode@multi
5862     \let\eql@numbering@active\eql@false}},%
5863   {{single,1}{\let\eql@numbering@mode\eql@numbering@mode@single
5864     \let\eql@numbering@active\eql@true}},%
5865   {{multi,@}{\let\eql@numbering@mode\eql@numbering@mode@multi
5866     \let\eql@numbering@active\eql@true}},%
5867   {\relax{\eql@numbering@set{#1}}}}
5868 \eql@define@key\eql@keycat{nonumber,nn,*}[]{%
5869   \let\eql@numbering@active\eql@false}
5870 \eql@define@key\eql@keycat{donumber,dn,!}[]{%
5871   \let\eql@numbering@active\eql@true}
5872 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
5873 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
5874 \eql@define@key\eql@keycat{tags,eqno}{%
5875   \eql@decide@select{#3}{#2}{#1}{%
5876     {\right,r}{\let\eql@tagsleft\eql@false}},%
5877     {\left,l}{\let\eql@tagsleft\eql@true}}}}
5878 \eql@define@key\eql@keycat{evadetag,avoidtag}[true]{%
5879   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}
5880 \eql@define@key\eql@keycat{tagbetween}[true]{%
5881   \eql@decide@bool{#3}{#2}{#1}\eql@tagpos@doconvert}

```

TODO: describe

```

5882 \eql@define@key{control}{nonumber,nn,*}[]{\global\@eqnswfalse}
5883 \eql@define@key{control}{donumber,dn,!}[]{\global\@eqnswtrue}
5884 \eql@define@key{control}{numberhere}[]{\eql@numberhere}
5885 \eql@define@key{control}{numbernext}[]{\eql@numbernext}

```

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

```

5886 \def\eql@keycat{equations,setup}
5887 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
5888   {{center,c}{\let\eql@layoutleft\eql@false}},%
5889   {{left,l}{\let\eql@layoutleft\eql@true}}}}
5890 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
5891 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
5892 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin{#1}}
5893 \eql@define@key\eql@keycat{leftmargin*}{%
5894   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin{\the\dimen@}}
5895 \eql@define@key\eql@keycat{minleftmargin}{%
5896   \def\eql@layoutleftmarginmin{#1}}
5897 \eql@define@key\eql@keycat{maxleftmargin}{%
5898   \eql@decide@select{#3}{#2}{#1}{%
5899     {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
5900     {\relax{\def\eql@layoutleftmarginmax{#1}}}}}
5901 \def\eql@keycat{equations,box}
5902 \eql@define@key\eql@keycat{margin}{%
5903   \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
5904 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}
5905 \eql@define@key\eql@keycat{marginright}{\def\eql@display@marginright{#1}}
5906 \eql@define@key{equations}{linewidth,width}{\def\eql@display@linewidth{#1}}

```

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

```

5907 \def\eql@keycat{equations,setup}
5908 \eql@define@key\eql@keycat{alignshrink}{\eql@decide@select{#3}{#2}{#1}{%
5909     {{max,full,4}{\eql@alignbadness@inf@bad}},%
5910     {{high,3}{\eql@alignbadness@54\relax}},%
5911     {{med,medium,2}{\eql@alignbadness@18\relax}},%
5912     {{low,1}{\eql@alignbadness@6\relax}},%
5913     {{0,\eql@decide@false}{\eql@alignbadness@z@}}}}
5914 \eql@define@key\eql@keycat{tagshrink}{\eql@decide@select{#3}{#2}{#1}{%
5915     {{max,full,4}{\eql@tagbadness@inf@bad}},%
5916     {{high,3}{\eql@tagbadness@54\relax}},%
5917     {{med,medium,2}{\eql@tagbadness@18\relax}},%
5918     {{low,1}{\eql@tagbadness@6\relax}},%
5919     {{0,\eql@decide@false}{\eql@tagbadness@z@}}}}
5920 \eql@define@key\eql@keycat{alignbadness}{\eql@alignbadness@numexpr#1\relax}
5921 \eql@define@key\eql@keycat{tagbadness}{\eql@tagbadness@numexpr#1\relax}
5922 \eql@define@key\eql@keycat{mincolsep}{\eql@decide@select{#3}{#2}{#1}{%
5923     {{0,\eql@decide@false}{\def\eql@colsepmin@val{0pt}}},%
5924     {\relax{\def\eql@colsepmin@val{#1}}}}}
5925 \eql@define@key\eql@keycat{maxcolsep}{\eql@decide@select{#3}{#2}{#1}{%
5926     {\eql@decide@false{\def\eql@colsepmax@val{.5\maxdimen}}},%
5927     {\relax{\def\eql@colsepmax@val{#1}}}}}
5928 \eql@define@key\eql@keycat{fulllength}[true]{%
5929     \eql@decide@bool{#3}{#2}{#1}\eql@columns@fulllength}

```

TODO: is boxcolsep vs breakcolsep okay??!

```

5930 \eql@define@key\eql@keycat{linesep}{\eql@decide@select{#3}{#2}{#1}{%
5931     {{0,\eql@decide@false}{\def\eql@break@line@sep{0pt}}},%
5932     {\relax{\def\eql@break@line@sep{#1}}}}}
5933 \eql@define@key\eql@keycat{linesep*}{\eql@decide@select{#3}{#2}{#1}{%
5934     {{0,\eql@decide@false}{\def\eql@break@line@shortsep{0pt}}},%
5935     {\relax{\def\eql@break@line@shortsep{#1}}}}}
5936 \eql@define@key{box,setup}{colsep}{\eql@decide@select{#3}{#2}{#1}{%
5937     {{0,\eql@decide@false}{\def\eql@box@colsep{0pt}}},%
5938     {{short}{\def\eql@box@colsep{\eql@box@shortsep}}},%
5939     {\relax{\def\eql@box@colsep{#1}}}}}
5940 \let\eql@break@col@sep\eql@box@colsep
5941 \eql@define@key{equations}{colsep}{\eql@decide@select{#3}{#2}{#1}{%
5942     {{0,\eql@decide@false}{\def\eql@break@col@sep{0pt}}},%
5943     {\relax{\def\eql@break@col@sep{#1}}}}}
5944 \let\eql@colsepmin@val\eql@box@colsep
5945 \let\eql@colsepmax@val\eql@box@colsep
5946 \let\eql@box@colsep\eql@break@col@sep
5947 \eql@define@key\eql@keycat{colsep*}{\eql@decide@select{#3}{#2}{#1}{%
5948     {{0,\eql@decide@false}{\def\eql@break@col@shortsep{0pt}}},%
5949     {\relax{\def\eql@break@col@shortsep{#1}}}}}
5950 \eql@define@key{box,setup}{colsep*}{\eql@decide@select{#3}{#2}{#1}{%
5951     {{0,\eql@decide@false}{\def\eql@box@shortsep{0pt}}},%
5952     {\relax{\def\eql@box@shortsep{#1}}}}}
5953 \eql@define@key{box,setup}{condsep}{\eql@decide@select{#3}{#2}{#1}{%
5954     {{0,\eql@decide@false}{\def\eql@box@condsep{0pt}}},%
5955     {\relax{\def\eql@box@condsep{#1}}}}}

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5956 \def\eql@keycat{equations,box,setup}

```



```

5957 \eqld@define@key\eqld@keycat{shape}[default]{\eqld@shape@set{#1}}
5958 \eqld@define@key\eqld@keycat{padding,pad}[indent]{%
5959   \eqld@decide@select{#3}{#2}{#1}{%
5960     {{max}}{\let\eqld@paddingleft@val\@undefined}},%
5961     {{indent}}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
5962     {{0,\eqld@decide@false}}{\def\eqld@paddingleft@val{0pt}}},%
5963     {\relax{\def\eqld@paddingleft@val{#1}}}}%
5964   \let\eqld@paddingright@val\eqld@paddingleft@val}
5965 \eqld@define@key\eqld@keycat{padleft}[indent]{%
5966   \eqld@decide@select{#3}{#2}{#1}{%
5967     {{max}}{\let\eqld@paddingleft@val\@undefined}},%
5968     {{indent}}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
5969     {{0,\eqld@decide@false}}{\def\eqld@paddingleft@val{0pt}}},%
5970     {\relax{\def\eqld@paddingleft@val{#1}}}}%
5971 \eqld@define@key\eqld@keycat{padright}[indent]{%
5972   \eqld@decide@select{#3}{#2}{#1}{%
5973     {{max}}{\let\eqld@paddingright@val\@undefined}},%
5974     {{indent}}{\def\eqld@paddingright@val{\eqld@indent@val}}},%
5975     {{0,\eqld@decide@false}}{\def\eqld@paddingright@val{0pt}}},%
5976     {\relax{\def\eqld@paddingright@val{#1}}}}%
5977 \eqld@define@key\eqld@keycat{indent}[2em]{%
5978   \def\eqld@indent@val{#1}}

```

TODO: describe

```

5979 \def\eqld@shape@rel{\eqncontrol{align=left}}
5980 \def\eqld@shape@cont{\eqncontrol{align=left,shift=*}}
5981 \eqld@define@key\eqld@keycat{shaperel}[]{\def\eqld@shape@rel{#1}}
5982 \eqld@define@key\eqld@keycat{shapecont}[]{\def\eqld@shape@cont{#1}}

```

TODO: describe

```

5983 \eqld@define@key{control}{align}[]{%
5984   \eqld@decide@select{#3}{#2}{#1}{%
5985     {{l,left}}{\global\eqld@append\eqld@cell@container{\eqld@shape@pos@z@}}},%
5986     {{c,center}}{\global\eqld@append\eqld@cell@container{\eqld@shape@pos@one@}}},%
5987     {{r,right}}{\global\eqld@append\eqld@cell@container{\eqld@shape@pos@two@}}}}%
5988 \eqld@define@key{control}{shift,shiftto}[]{%
5989   \eqld@decide@select{#3}{#2}{#1}{%
5990     {{*,indent}}{\eqld@shape@alignamount@set{\eqld@indent@}}},%
5991     {{!,outdent}}{\eqld@shape@alignamount@set{-\eqld@indent@}}},%
5992     {\relax{\eqld@shape@alignamount@set{#1}}}}%
5993 \eqld@define@key{control}{shift*,shiftby}[]{\eqld@shape@alignamount@add{#1}}

```

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

```

5994 \def\eqld@keycat{equations,box,setup}
5995 \eqld@define@key\eqld@keycat{classout}{\eqld@class@innerleft@set{#1}}
5996 \eqld@define@key\eqld@keycat{classin}{\eqld@class@innerright@set{#1}}
5997 \eqld@define@key\eqld@keycat{classlead,classin*}{\eqld@class@innerlead@set{#1}}
5998 \eqld@define@key\eqld@keycat{rel}{\def\eqld@class@rel@symb{#1}}
5999 \eqld@define@key\eqld@keycat{classcont}{\def\eqld@class@rel@cont##1{#1}}
6000 \eqld@define@key\eqld@keycat{classstart}{\def\eqld@class@rel@start##1{#1}}
6001 \eqld@define@key{control}{rel}[\eqld@class@rel@symb]{\eqld@class@rel@make{#1}}
6002 \eqld@define@key{control}{rel;}[]{\eqld@class@rel@make{}}
6003 \eqld@define@key{control}{rel*}[]{\eqld@class@rel@make{}}
6004 \eqld@define@key\eqld@keycat{ampeq}[]{\eqld@class@ampeq}
6005 \eqld@define@key\eqld@keycat{eqamp}[]{\eqld@class@eqamp}
6006 \eqld@define@key\eqld@keycat{class}{\eqld@decide@select{#3}{#2}{#1}{%

```

```

6007  {\ampeq,amprel,eqafter,beforerel}\eq@class@ampeq},%
6008  {\eqamp,relamp,eqbefore,afterrel}\eq@class@eqamp}}}
```

Math Styles. Configure math classes at alignment marker:

```

6009 \eq@define@key\eq@keycat{style}[display]{%
6010   \eq@decide@select{#3}{#2}{#1}{%
6011     {text{\let\eq@mathstyle\empty}},%
6012     {display{\let\eq@mathstyle\displaystyle}}}}
6013 \eq@define@key{setup}{casesstyle}[display]{%
6014   \eq@decide@select{#3}{#2}{#1}{%
6015     {\eq@decide@false{\let\eq@cases@mathstyle\eq@false}},%
6016     {\text{\let\eq@cases@mathstyle\empty}},%
6017     {\display{\let\eq@cases@mathstyle\displaystyle}}}}
```

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

```

6018 \def\eq@punct@all#1#2#3#4#5\eq@punct@end{%
6019   \def\eq@tmp{#4}\def\eq@tmpa{1}%
6020   \ifx\eq@tmp\eq@tmpa
6021     \ifnum#5=1111\relax
6022       \eq@punct@set\eq@punct@col{#1}%
6023       \eq@punct@set\eq@punct@line{#2}%
6024       \eq@punct@set\eq@punct@block{#3}%
6025     \else\ifnum#5=111\relax
6026       \eq@punct@set\eq@punct@line{#1}%
6027       \eq@punct@set\eq@punct@block{#2}%
6028     \else\ifnum#5=11\relax
6029       \eq@punct@set\eq@punct@block{#1}%
6030     \else
6031       \eq@punct@clear
6032     \fi\fi\fi
6033   \else
6034     \eq@error{Too many arguments to punctall}%
6035   \fi
6036 }
```

TODO: describe

```

6037 \def\eq@keycat{equations,box,setup}
6038 \eq@define@key\eq@keycat{punctsep}[\,]{\def\eq@punct@sep{#1}}
6039 \eq@define@key\eq@keycat{punctclass}[\mathclose{}]{\def\eq@punct@class{#1}}
6040 \eq@define@key\eq@keycat{punct}[.]{\eq@punct@set\eq@punct@block{#1}}
6041 \eq@define@key\eq@keycat{punct*}[]{\eq@punct@set\eq@punct@block\relax}
6042 \eq@define@key\eq@keycat{punctline}[,]{\eq@punct@set\eq@punct@line{#1}}
6043 \eq@define@key\eq@keycat{punctline*}[]{\eq@punct@set\eq@punct@line\relax}
6044 \eq@define@key\eq@keycat{punctcol}[,]{\eq@punct@set\eq@punct@col{#1}}
6045 \eq@define@key\eq@keycat{punctcol*}[]{\eq@punct@set\eq@punct@col\relax}
6046 \eq@define@key\eq@keycat{punctall}[,,]{\eq@punct@all#111111\eq@punct@end}
6047 \eq@define@key{box}{punctterm}[true]{%
6048   \eq@decide@bool{#3}{#2}{#1}\eq@box@punct@term}

6049 \eq@define@key{control}{punctsep}[\,]{\def\eq@punct@sep{#1}}
6050 \eq@define@key{control}{setpunct}[.]{\eq@punct@set\eq@punct@next{#1}}
6051 \eq@define@key{control}{setpunct}[,]{\eq@punct@set\eq@punct@next{#1}}
6052 \eq@define@key{control}{setpunct*}[]{\let\eq@punct@next\relax}
6053 \eq@define@key{control}{punct,punctapply}[\relax]{%
6054   \eq@punct@set\eq@punct@next{#1}\eq@punct@apply@top}
6055 \eq@define@key{control}{punctline}[]{\eq@punct@print@line}
```



```
6056 \eqld@define@key{control}{punctcol}[]{\eqlpunct@print@col}
```

Frames. **TODO:** describe

```
6057 \eqld@define@key{box}{frame}[\fbox]{%
6058   \def\eqld@box@frame#1{%
6059     \ifx\eqld@box@frame\empty\let\eqld@box@frame\@firstofone\fi}
6060 \eqld@define@key{box}{wrap}{\eqld@box@wrap#1}
6061 \eqld@define@key{box}{delim}[r]{\eqld@decide@delim{#3}{#2}{#1}}
6062 \eqld@define@key{box}{ldelim}{\eqld@box@ldelim#1}
6063 \eqld@define@key{box}{rdelim}{\eqld@box@rdelim#1}
6064 \eqld@define@key{box}{lbrace}[]{\eqld@box@ldelim\lbrace}
6065 \eqld@define@key{box}{rbrace}[]{\eqld@box@rdelim\rbrace}
6066 \eqld@define@key{box}{lbrace,lbraces}[]{\eqld@box@delim\lbrace\rbrace}
6067 \eqld@define@key\eqld@keycat{braces}[lr]{%
6068   \eqld@decide@select{#3}{#2}{#1}{%
6069     {\eqld@decide@false}{\eqld@box@wrap}{}}},%
6070     {\l,left}{\eqld@box@ldelim\lbrace}},%
6071     {\r,right}{\eqld@box@rdelim\rbrace}},%
6072     {\eqld@decide@true,lr,both}{\eqld@box@delim\lbrace\rbrace}}}
```

TODO: describe

```
6073 \eqld@define@key{control}{framecell}[\fbox]{%
6074   \global\eqld@append\eqld@cell@container{\def\eqld@frame@cmd{#1}}}
6075 \eqld@define@key{control}{frametag}[\fbox]{%
6076   \global\eqld@append\eqld@tags@container{\def\eqld@tags@frame@cmd{#1}}}
```

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

```
6077 \eqld@define@key{equations,box}{alt}{}
```

Injectsions.

```
6078 \eqld@define@key{control}{inject}{%
6079   \global\eqld@append\eqld@interline@container{%
6080     \eqld@append\eqld@display@injectbefore{#1}}}
6081 \eqld@define@key{control}{inject*}{%
6082   \global\eqld@append\eqld@interline@container{%
6083     \eqld@append\eqld@display@injectafter{#1}}}
6084 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
6085 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
6086 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
6087 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}
```

TODO: describe

```
6088 \eqld@define@key{markline}{pos}{%
6089   \eqld@decide@select{#3}{#2}{#1}{%
6090     {\below,push}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
6091     {\baseline}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
6092     {\bottom}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}
6093 \eqld@define@key{markline}{below,push}[]{%
6094   \let\eqld@markline@pos\eqld@markline@pos@below}
6095 \eqld@define@key{markline}{baseline}[]{%
6096   \let\eqld@markline@pos\eqld@markline@pos@baseline}
6097 \eqld@define@key{markline}{bottom}[]{%
```

```

6098 \let\eql@markline@pos\eql@markline@pos@bottom}
6099 \eql@define@key{markline}{shift}{\def\eql@markline@shift{#1}}
6100 \eql@define@key{markline}{symbol}{\def\eql@markline@symbol{#1}}
6101 \eql@define@key{markline}{qed}[]{\let\eql@markline@symbol\eql@markline@qed}
6102 \eql@define@key{setup}{marksymbol}{\def\eql@markline@symbol{#1}}
6103 \eql@define@key{setup}{qedsymbol}{\def\eql@markline@qed{#1}}
6104 \eql@define@key{setup}{markpos}{%
6105 \eql@decide@select{#3}{#2}{#1}{%
6106   {{below}}{\let\eql@markline@pos\eql@markline@pos@below}},%
6107   {{baseline}}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
6108   {{bottom}}{\let\eql@markline@pos\eql@markline@pos@bottom}}}}

```

Global Switches. Set global switches:

```

6109 \let\eql@multi@linesfallback\eql@false
6110 \let\eql@scan@par\eql@false
6111 \let\eql@single@cr@mode\eql@false
6112 \let\eql@amp@mode\eql@true
6113 \let\eql@ampproof@active\eql@false
6114 \let\eql@parseopt@warn@env\eql@warn@parseopt
6115 \let\eql@parseopt@warn@cr\@empty

6116 \eql@define@key{equations,setup}{linesfallback}[true]{%
6117 \eql@decide@select{#3}{#2}{#1}{%
6118   {{\eql@decide@false{\let\eql@multi@linesfallback\eql@false}}},%
6119   {{reuse,lean}}{\let\eql@multi@linesfallback\z@}},%
6120   {{measure,full,\eql@decide@true}}{\let\eql@multi@linesfallback\eql@true}}}}
6121 \eql@define@key{setup}{ampproof}[true]{%
6122 \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
6123 \eql@define@key{equations,setup}{equationcr}[true]{%
6124 \eql@decide@select{#3}{#2}{#1}{%
6125   {{\eql@decide@false{\let\eql@single@cr@mode\eql@false}}},%
6126   {{\eql@decide@true,break}}{\let\eql@single@cr@mode\eql@break@cr}},%
6127   {{error,verbose}}{\let\eql@single@cr@mode\eql@single@cr@error}}}}
6128 \eql@define@key{equations,box,setup}{amp}[true]{%
6129 \eql@decide@bool{#3}{#2}{#1}\eql@amp@mode}
6130 \eql@define@key{equations,box,setup}{rescan}[true]{%
6131 \eql@decide@if{#3}{#2}{#1}%
6132   {\let\eql@scan@body\eql@scan@body@rescan}%
6133   {\let\eql@scan@body\eql@scan@body@dump}}
6134 \eql@define@key{equations,box,setup}{scanpar}[true]{%
6135 \eql@decide@bool{#3}{#2}{#1}\eql@scan@par}
6136 \eql@define@key{setup}{defaults}{%
6137 \eql@decide@select{#3}{#2}{#1}{%
6138   {{classic}}{\eql@defaults@classic}},%
6139   {{eqnlines}}{\eql@defaults@eqnlines}}}}
6140 \eql@define@key{equations,box,setup}{verbose}[true]{%
6141 \eql@decide@if{#3}{#2}{#1}\eql@verbose@on\eql@verbose@off}

```

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

```

6142 \let\eql@provide@opt@env\tw@
6143 \let\eql@provide@opt@amsmathpatch\eql@false
6144 \let\eql@provide@opt@backup\eql@false
6145 \let\eql@provide@opt@ang\eql@true
6146 \let\eql@provide@opt@eqref\eql@true
6147 \let\eql@provide@opt@matrix\eql@true

```

```

6148 \eql@define@key{setup}{amsmathends,amsmathpatch}[true]{%
6149   \eql@error@packageoption{#2}%
6150   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathpatch}
6151 \eql@define@key{setup}{backup}[true]{%
6152   \eql@error@packageoption{#2}%
6153   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
6154 \eql@define@key{setup}{env}[equation]{%
6155   \eql@error@packageoption{#2}%
6156   \eql@decide@select{#3}{#2}{#1}{%
6157     {none,\eql@decide@false}{\let\eql@provide@opt@env\z},%
6158     {equation,latex}{\let\eql@provide@opt@env\one}},%
6159     {{amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw}}}}
6160 \eql@define@key{setup}{ang}[true]{%
6161   \eql@error@packageoption{#2}%
6162   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
6163 \eql@define@key{setup}{eqref}[true]{%
6164   \eql@error@packageoption{#2}%
6165   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}
6166 \eql@define@key{setup}{matrix}[true]{%
6167   \eql@error@packageoption{#2}%
6168   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@matrix}

```

Shortcut Options. **TODO:** describe

```

6169 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
6170 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
6171 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
6172 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
6173 \def\eql@parseopt@eqamp#1{\eqnaddopt{eqamp}\eql@parseopt@peek}
6174 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
6175 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
6176 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
6177 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
6178 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
6179 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
6180 \def\eql@parseopt@punctclear#1{\eql@parseopt@peek'~}
6181 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
6182 \def\eql@parseopt@punctnext#1#2{%
6183   \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
6184 \def\eql@parseopt@punctblock#1#2{%
6185   \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
6186 \def\eql@parseopt@punctterm#1{\let\eql@punct@term\eql@true\eql@parseopt@peek}
6187 \def\eql@parseopt@rel symb#1{\eql@parseopt@peek|\eql@class@rel@symb}
6188 \def\eql@parseopt@rel cont#1{\eql@parseopt@peek|{}}
6189 \def\eql@parseopt@rel start#1{\eql@parseopt@peek|\relax}
6190 \def\eql@parseopt@vspace[#1]{%
6191   \advance\eql@vspaceskip@glueexpr#1\relax\eql@parseopt@peek}

```

16.3 Parameter Presets

The package offers two parameter presets which lead to somewhat different layout. Instead of setting the internal parameters directly, we expose them as public settings so that they are easier to read and such that individual settings can be used to compose own layouts.

`\eql@defaults@classic` The preset `classic` aims to reproduce the $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

6192 \def\eql@defaults@classic{%
6193   \eqnlineset{numberline=all}%
6194   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
6195   \eqnlineset{maxcolsep=off}%
6196   \eqnlineset{spread={\jot}}%
6197   \eqnlineset{tagmargin}%
6198   \eqnlineset{tagmarginratio=1}%
6199   \eqnlineset{tagmarginthreshold=0.5}%
6200   \eqnlineset{leftmargin={\leftmargini}}%
6201   \eqnlineset{padding=max}%
6202   \eqnlineset{evadetag=off}%
6203   \eqnlineset{displayheight=off}%
6204   \eqnlineset{displaydepth=off}%
6205   \eqnlineset{shortmode=belowsingle}%
6206   \eqnlineset{abovecontmode=short}%
6207   \eqnlineset{belowcontmode=short}%
6208   \eqnlineset{aboveparmode=long}%
6209   \eqnlineset{belowparmode=long}%
6210   \eqnlineset{abovetopmode=long}%
6211   \eqnlineset{belowtopmode=long}%
6212   \eqnlineset{abovelongskip={\abovedisplayskip}}%
6213   \eqnlineset{belowlongskip={\belowdisplayskip}}%
6214   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
6215   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
6216   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
6217   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
6218   \eqnlineset{abovecontskip=0pt}%
6219   \eqnlineset{belowcontskip=0pt}%
6220   \eqnlineset{aboveparskip=0pt}%
6221   \eqnlineset{belowparskip=0pt}%
6222   \eqnlineset{abovetopskip=0pt}%
6223   \eqnlineset{belowtopskip=0pt}%
6224   \eqnlineset{abovetagskip=0pt}%
6225   \eqnlineset{belowtagskip=0pt}%
6226   \eqnlineset{allowbreaks=0}%
6227   \eqnlineset{equationcr=off}%
6228   \eqnlineset{amp=off}%
6229   \eqnlineset{modifier=off}%
6230   \eqnlineset{linesfallback=false}%
6231   \eqnlineset{casesstyle=text}%
6232   \eqnlineset{sqropt=nonumber}%
6233   \eqnlineset{angopt=nonumber}%
6234 }

```

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 12pt computer modern font such that 10pt fonts will lead to slightly reduced spacing. Apart from that, the **eqnlines** setting makes some deliberate layout choices that deviate significantly from **classic** (maximum column separation, no shortening below equations):

```

6235 \def\eql@defaults@eqnlines{%
6236   \eqnlineset{numberline=all}%
6237   \eqnlineset{mintagsep=.5em}%
6238   \eqnlineset{maxcolsep=2em}%
6239   \eqnlineset{spread={0.2\normalbaselineskip}}%
6240   \eqnlineset{tagmargin}%

```

```

6241 \eqnlineset{tagmarginratio=.334}%
6242 \eqnlineset{tagmarginthreshold=0.5}%
6243 \eqnlineset{leftmargin={\leftmargini}}%
6244 \eqnlineset{padding=0pt}%
6245 \eqnlineset{evadetag}%
6246 \eqnlineset{displayheight=strut}%
6247 \eqnlineset{displaydepth=strut}%
6248 \eqnlineset{shortmode=above}%
6249 \eqnlineset{abovecontmode=noskip}%
6250 \eqnlineset{belowcontmode=long}%
6251 \eqnlineset{aboveparmode=long}%
6252 \eqnlineset{belowparmode=long}%
6253 \eqnlineset{abovetopmode=noskip}%
6254 \eqnlineset{belowtopmode=long}%
6255 \eqnlineset{longskip={0.75\normalbaselineskip
6256   plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
6257 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
6258   plus 0.25\normalbaselineskip}}%
6259 \eqnlineset{belowshortskip={0.0\normalbaselineskip
6260   plus 0.25\normalbaselineskip}}%
6261 \eqnlineset{medskip={0.4\normalbaselineskip
6262   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
6263 \eqnlineset{abovecontskip=0pt}%
6264 \eqnlineset{belowcontskip=0pt}%
6265 \eqnlineset{aboveparskip=0pt}%
6266 \eqnlineset{belowparskip=0pt}%
6267 \eqnlineset{abovetopskip=0pt}%
6268 \eqnlineset{belowtopskip=0pt}%
6269 \eqnlineset{abovetagskip={0.25\normalbaselineskip
6270   minus 0.25\normalbaselineskip}}%
6271 \eqnlineset{belowtagskip={0.25\normalbaselineskip
6272   minus 0.25\normalbaselineskip}}%
6273 \eqnlineset{allowbreaks=3}%
6274 \eqnlineset{equationcr=break}%
6275 \eqnlineset{amp=on}%
6276 \eqnlineset{modifier=on,modifierenv=off,modifierbreak=off}%
6277 \eqnlineset{linesfallback=true}%
6278 \eqnlineset{casesstyle=false}%
6279 \eqnlineset{sqropt}%
6280 \eqnlineset{angopt}%
6281 }

```

16.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of \LaTeX and `amsmath` carefully.

Tools.

```

\eq1@provide@movecmd We introduce a couple of tools to rename and undefine commands and environments:
\eq1@provide@moveenv
@provide@undefinecmd 6282 \def\eq1@provide@movecmd#1#2{%
@provide@undefineenv 6283   \eq1@letcs{#1\expandafter}\csname#2\endcsname
6284 }
6285 \def\eq1@provide@moveenv#1#2{%
6286   \eq1@provide@movecmd{#1}{#2}%
6287   \ifcsname end#2\endcsname

```

```

6288 \eql@provide@movecmd{end#1}{end#2}%
6289 \fi
6290 }
6291 \def\eql@provide@undefinecmd#1{%
6292 \eql@letcs{#1}\@undefined
6293 }
6294 \def\eql@provide@undefineenv#1{%
6295 \eql@provide@undefinecmd{#1}%
6296 \eql@provide@undefinecmd{end#1}%
6297 }

```

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:

```

6298 \def\eql@amsmath@endfix#1#2{%
6299 \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
6300 \expandafter\ifx\csname end#1\endcsname\eql@tmpa
6301 \eql@provide@movecmd{end#1}{end#2}%
6302 \fi
6303 }

```

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

```

6304 \def\eql@amsmath@fixmatrix#1{%
6305 \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
6306 \begingroup
6307 \let\matrix@check\@gobble
6308 \def\env@matrix{\noexpand\env@matrix}%
6309 \def\env@cases{\noexpand\env@cases}%
6310 \global\edef\eql@tmp{\eql@tmp}%
6311 \endgroup
6312 \eql@letcs{#1}\eql@tmp
6313 }
6314 \def\eql@amsmath@fixmatrixend#1{%
6315 \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
6316 \begingroup
6317 \expandafter\def\expandafter\endmatrix\expandafter{%
6318 \expandafter\unexpanded\expandafter{\endmatrix}}%
6319 \global\long\edef\eql@tmp{\eql@tmp}%
6320 \endgroup
6321 \eql@letcs{end#1}\eql@tmp
6322 }

```

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

```

6323 \def\eql@amsmath@fixends{%
6324 \eql@amsmath@after{%
6325 \eql@amsmath@endfix{flalign}{align}%
6326 \eql@amsmath@endfix{alignat}{align}%

```

```

6327 \eq@amsmath@endfix{xalignat}{align}%
6328 \eq@amsmath@endfix{xxalignat}{align}%
6329 \eq@amsmath@endfix{gather*}{gather}%
6330 \eq@amsmath@endfix{multline*}{multline}%
6331 \eq@amsmath@endfix{align*}{align}%
6332 \eq@amsmath@endfix{flalign*}{align}%
6333 \eq@amsmath@endfix{alignat*}{align}%
6334 \eq@amsmath@endfix{xalignat*}{align}%
6335 \eq@amsmath@endfix{gathered}{aligned}%
6336 \eq@amsmath@endfix{alignedat}{aligned}%
6337 }
6338 }

```

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

```

6339 \def\eq@amsmath@fixmatrices{%
6340 \eq@amsmath@after{%
6341 \eq@amsmath@fixmatrix{cases}%
6342 \eq@amsmath@fixmatrix{matrix}%
6343 \eq@amsmath@fixmatrix{pmatrix}%
6344 \eq@amsmath@fixmatrixend{pmatrix}%
6345 \eq@amsmath@fixmatrixend{bmatrix}%
6346 \eq@amsmath@fixmatrixend{Bmatrix}%
6347 \eq@amsmath@fixmatrixend{vmatrix}%
6348 \eq@amsmath@fixmatrixend{Vmatrix}%
6349 }
6350 }

```

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

```

6351 \def\eq@provide@backup@amsenv#1{%
6352 \eq@amsmath@after{%
6353 \eq@provide@moveenv{ams#1}{#1}%
6354 \eq@tagging@register@luamml{ams#1}%
6355 \eq@markline@amsthm@move{ams#1}{#1}%
6356 }%
6357 }

```

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

```

6358 \def\eq@provide@backup@amsbox#1{%
6359 \eq@amsmath@after{%
6360 \eq@provide@moveenv{ams#1}{#1}%
6361 }%
6362 }

```

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

```

6363 \def\eq@provide@backup@eqref{%
6364 \eq@amsmath@after{%
6365 \eq@provide@movecmd{amseqref}{eqref}%
6366 }%
6367 }

```

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

```
6368 \def\eql@provide@backup@multlined{%
6369   \AddToHook{package/mathtools/after}{%
6370     \eql@provide@moveenv{amsmultlined}{multlined}}%
6371 }
```

vide@backup@equation The L^AT_EX 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 L^AT_EX and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original L^AT_EX `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:

```
6372 \def\eql@provide@backup@equation{%
6373   \eql@amsmath@after{%
6374     \eql@provide@moveenv{amsequeation}{equation}%
6375     \eql@provide@moveenv{amsequeation*}{equation*}%
6376     \eql@tagging@register@env{amsequeation}%
6377     \eql@tagging@register@env{amsequeation*}%
6378     \eql@tagging@register@luamml{amsequeation}%
6379     \eql@tagging@register@luamml{amsequeation*}%
6380     \eql@markline@amsthm@move{amsequeation}{equation}%
6381     \eql@markline@amsthm@move{amsequeation*}{equation*}%
6382   }%
6383   \AddToHook{package/hyperref/after}{%
6384     \@ifpackageloaded{amsmath}{}%
6385       \eql@provide@moveenv{hyperrefequation}{equation}%
6386       \eql@tagging@register@env{hyperrefequation}%
6387       \eql@tagging@register@luamml{hyperrefequation}%
6388       \eql@markline@amsthm@move{hyperequation}{equation}%
6389     }%
6390   }%
6391   \@ifpackageloaded{amsmath}{}%
6392     \@ifpackageloaded{hyperref}{%
6393       \let\latexequation\H@equation
6394       \let\endlatexequation\H@endequation
6395     }{\eql@provide@moveenv{latexequation}{equation}}%
6396     \eql@tagging@register@env{latexequation}%
6397     \eql@tagging@register@luamml{latexequation}%
6398     \eql@markline@amsthm@move{latexequation}{equation}%
6399   }%
6400 }
```

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

```
6401 \def\eql@provide@backup@displaymath{%
6402   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
6403   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
6404 }
```

@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:


```

6405 \def\eql@provide@backup@subequations{%
6406   \eql@amsmath@after{%
6407     \eql@provide@moveenv{amssubequations}{subequations}%
6408   }%
6409   \AddToHook{package/hyperref/after}{%
6410     \AddToHook{cmd/amssubequations/before}{%
6411       {%
6412         \stepcounter{equation}%
6413         \protected@edef\theHparentequation{\theHequation}%
6414         \addtocounter{equation}{-1}%
6415       }%
6416       \AddToHook{cmd/amssubequations/after}{%
6417         {%
6418           \def\theHequation{\theHparentequation\alph{equation}}%
6419           \ignorespaces
6420         }%
6421       }%
6422 }

```

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

```

6423 \def\eql@provide@backup{%
6424   \eql@provide@backup@eqref
6425   \eql@provide@backup@equation
6426   \eql@provide@backup@displaymath
6427   \eql@provide@backup@amsenv{gather}%
6428   \eql@provide@backup@amsenv{multline}%
6429   \eql@provide@backup@amsenv{align}%
6430   \eql@provide@backup@amsenv{flalign}%
6431   \eql@provide@backup@amsenv{alignat}%
6432   \eql@provide@backup@amsenv{xalignat}%
6433   \eql@provide@backup@amsenv{xxalignat}%
6434   \eql@provide@backup@amsenv{gather*}%
6435   \eql@provide@backup@amsenv{multline*}%
6436   \eql@provide@backup@amsenv{align*}%
6437   \eql@provide@backup@amsenv{flalign*}%
6438   \eql@provide@backup@amsenv{alignat*}%
6439   \eql@provide@backup@amsenv{xalignat*}%
6440   \eql@provide@backup@amsbox{gathered}%
6441   \eql@provide@backup@multlined
6442   \eql@provide@backup@amsbox{aligned}%
6443   \eql@provide@backup@amsbox{alignedat}%
6444   \eql@provide@backup@amsbox{cases}%
6445   \eql@provide@backup@amsbox{matrix}%
6446   \eql@provide@backup@amsbox{pmatrix}%
6447   \eql@provide@backup@amsbox{bmatrix}%
6448   \eql@provide@backup@amsbox{Bmatrix}%
6449   \eql@provide@backup@amsbox{vmatrix}%
6450   \eql@provide@backup@amsbox{Vmatrix}%
6451   \eql@provide@backup@subequations
6452 }

```

Replacement amsmath Environments. TODO: describe

```

6453 \def\eql@alignat@gobblecol#1{%
6454   \eql@ifnextchar@tight\bgroup{\@firstoftwo{#1}}{#1}}

```

`\eql@gathered` (*env.*) Define replacement versions for boxed environments `gathered`, `multlined` and `aligned`
`\eql@multlined` (*env.*)
`\eql@aligned` (*env.*)

which forward to equationsbox with specific presets:

```

6455 \newenvironment{eql@gathered}
6456   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
6457 \newenvironment{eql@multlined}
6458   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
6459 \newenvironment{eql@aligned}
6460   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
6461 \newenvironment{eql@alignedat}
6462   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
6463   {\endequationsbox}
6464 \newenvironment{eql@cases}
6465   {\eqnaddopt{cases}\equationsbox}{\endequationsbox}
6466 \newenvironment{eql@matrix}
6467   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=.}\equationsbox}
6468   {\endequationsbox}
6469 \newenvironment{eql@pmatrix}
6470   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=r}\equationsbox}
6471   {\endequationsbox}
6472 \newenvironment{eql@bmatrix}
6473   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=s}\equationsbox}
6474   {\endequationsbox}
6475 \newenvironment{eql@Bmatrix}
6476   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=c}\equationsbox}
6477   {\endequationsbox}
6478 \newenvironment{eql@vmatrix}
6479   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=v}\equationsbox}
6480   {\endequationsbox}
6481 \newenvironment{eql@Vmatrix}
6482   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=d}\equationsbox}
6483   {\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.*)

```

6484 \newenvironment{eql@equation}
6485   {\eqnaddopt{equation}\equations}{\endequations}
6486 \newenvironment{eql@displaymath}
6487   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
6488 \newenvironment{eql@gather}
6489   {\eqnaddopt{lines}\equations}{\endequations}
6490 \newenvironment{eql@multline}
6491   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
6492   {\endequations}
6493 \newenvironment{eql@align}
6494   {\eqnaddopt{columns}\equations}{\endequations}
6495 \newenvironment{eql@flalign}
6496   {\eqnaddopt{fulllength}\eql@align}{\endequations}
6497 \newenvironment{eql@alignat}
6498   {\eqnaddopt{colsep=off}\eql@xalignat}{\endequations}
6499 \newenvironment{eql@xalignat}
6500   {\eql@alignat@gobblecol\eql@align}{\endequations}
6501 \newenvironment{eql@xxalignat}
6502   {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
6503 \newenvironment{eql@equation*}
6504   {\eqnaddopt{nonumber}\eql@equation}{\endequations}
6505 \newenvironment{eql@gather*}
6506   {\eqnaddopt{nonumber}\eql@gather}{\endequations}

```

```

6507 \newenvironment{eql@multline*}
6508   {\eqnaddopt{nonumber}\eql@multline}{\endequations}
6509 \newenvironment{eql@align*}
6510   {\eqnaddopt{nonumber}\eql@align}{\endequations}
6511 \newenvironment{eql@flalign*}
6512   {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
6513 \newenvironment{eql@alignat*}
6514   {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
6515 \newenvironment{eql@xalignat*}
6516   {\eqnaddopt{nonumber}\eql@xalignat}{\endequations}

```

Install Additional Environments. The additional environments need to be installed at their intended names which can be adjusted by the user.

`eql@provide@onlyonce` Process arguments for providing a specific environment. #1 describes the environment using the `amsmath` name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the `amsmath` environment in place making sure that the replacement is robust against loading `amsmath` before or after. If #2 equals ‘*’, just overwrite the `amsmath` environment in place immediately (e.g. within a block in the document body):

```

6517 \def\eql@provide@onlyonce#1#2{%
6518   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
6519   \ifx\eql@tmp\eql@tmpa
6520     \let\eql@tmp\@empty
6521   \fi
6522   \ifx\eql@tmp\@empty
6523     \let\eql@tmp\@undefined
6524     \ifx\@nodocument\relax
6525       \def\eql@tmp{#1}%
6526     \fi
6527     \ifcsname eql@provided@#1\endcsname
6528       \def\eql@tmp{#1}%
6529     \fi
6530     \eql@letcs{eql@provided@#1}\eql@true
6531   \else
6532     \def\eql@tmpa{*}%
6533     \ifx\eql@tmp\eql@tmpa
6534       \def\eql@tmp{#1}%
6535     \fi
6536   \fi
6537 }

```

`\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:

```

6538 \def\eql@provide@eqref#1{%
6539   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
6540   \ifx\eql@tmp\eql@tmpa
6541     \let\eql@tmp\@empty
6542   \fi
6543   \ifx\eql@tmp\@empty
6544     \eql@provide@onlyonce{eqref}{}%
6545   \else
6546     \def\eql@tmpa{*}%
6547     \ifx\eql@tmp\eql@tmpa

```

```

6548     \def\eql@tmp{eqref}%
6549     \else
6550     \edef\eql@tmp{\expandafter\@gobble\string#1}%
6551     \fi
6552 \fi
6553 \ifdefined\eql@tmp
6554     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
6555 \else
6556     \eql@amsmath@after{%
6557         \eql@provide@movecmd{eqref}{eql@eqref}%
6558     }%
6559     \AddToHook{package/mathtools/after}{%
6560         \eql@provide@movecmd{eqref}{eql@eqref}%
6561     }%
6562     \eql@provide@movecmd{eqref}{eql@eqref}%
6563     \eql@amsmath@undefine\eqref
6564 \fi
6565 }

```

`\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:

```

6566 \def\eql@provide@amsenv#1#2{%
6567     \eql@provide@onlyonce{#1}{#2}%
6568     \ifdefined\eql@tmp
6569         \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6570         \eql@tagging@register@luamml{\eql@tmp}%
6571         \eql@markline@amsthm@register{\eql@tmp}%
6572     \else
6573         \eql@amsmath@after{%
6574             \eql@provide@moveenv{#1}{eql@#1}%
6575             \eql@markline@amsthm@register{#1}%
6576         }%
6577         \AddToHook{package/mathtools/after}{%
6578             \eql@provide@moveenv{#1}{eql@#1}%
6579             \eql@markline@amsthm@register{#1}%
6580         }%
6581         \eql@provide@moveenv{#1}{eql@#1}%
6582         \eql@markline@amsthm@register{#1}%
6583         \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6584     \fi
6585 }

```

`\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:

```

6586 \def\eql@provide@amsbox#1#2{%
6587     \eql@provide@onlyonce{#1}{#2}%
6588     \ifdefined\eql@tmp
6589         \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6590     \else
6591         \eql@amsmath@after{%
6592             \eql@provide@moveenv{#1}{eql@#1}}%
6593         \AddToHook{package/mathtools/after}{%
6594             \eql@provide@moveenv{#1}{eql@#1}}%
6595         \eql@provide@moveenv{#1}{eql@#1}%
6596         \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6597     \fi

```

6598 }

`\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:

```

6599 \def\eql@provide@multlined#1{%
6600   \eql@provide@onlyonce{multlined}{#1}%
6601   \ifdefined\eql@tmp
6602     \eql@provide@moveenv{\eql@tmp}{eql@multlined}%
6603   \else
6604     \AddToHook{package/mathtools/after}{%
6605       \eql@provide@moveenv{multlined}{eql@multlined}}%
6606     \eql@provide@moveenv{multlined}{eql@multlined}%
6607     \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
6608       \eql@provide@undefineenv{multlined}}}%
6609   \fi
6610 }
```

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

```

6611 \def\eql@provide@matrix#1#2#3{%
6612   \eql@provide@onlyonce{#1}{#3}%
6613   \ifdefined\eql@tmp
6614     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6615     \eql@tagging@register@luamml{\eql@tmp}%
6616   \else
6617     \eql@amsmath@after{%
6618       \eql@provide@moveenv{#1}{eql@#1}%
6619     }%
6620     \eql@provide@moveenv{#1}{eql@#1}%
6621     \ifdefined#2\eql@amsmath@before{\eql@provide@undefineenv{#1}}\fi%
6622   \fi
6623 }
```

`\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:

```

6624 \def\eql@provide@equation#1{%
6625   \eql@provide@onlyonce{equation}{#1}%
6626   \ifdefined\eql@tmp
6627     \eql@provide@moveenv{\eql@tmp}{eql@equation}%
6628     \eql@tagging@register@luamml{\eql@tmp}%
6629     \eql@markline@amsthm@register{\eql@tmp}%
6630   \else
6631     \eql@amsmath@after{%
6632       \eql@provide@moveenv{equation}{eql@equation}%
6633       \eql@markline@amsthm@register{equation}%
6634     }%
6635     \AddToHook{package/hyperref/after}{%
6636       \ifpackageloaded{amsmath}{%
6637         \eql@provide@moveenv{equation}{eql@equation}%
6638         \eql@markline@amsthm@register{equation}%
6639       }%
6640     }%
6641     \eql@provide@moveenv{equation}{eql@equation}%

```

```

6642 \eql@markline@amsthm@register{equation}%
6643 \ifdefined\eql@tagging@on
6644 \AddToHook{begindocument/end}{%
6645 \eql@provide@moveenv{equation}{eql@equation}%
6646 \eql@markline@amsthm@register{equation}%
6647 }%
6648 \fi
6649 \fi
6650 }

```

`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:

```

6651 \def\eql@provide@equationstar#1{%
6652 \eql@provide@onlyonce{equation*}{#1}%
6653 \ifdefined\eql@tmp
6654 \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
6655 \eql@tagging@register@luamml{\eql@tmp}%
6656 \eql@markline@amsthm@register{\eql@tmp}%
6657 \else
6658 \eql@amsmath@after{%
6659 \eql@provide@moveenv{equation*}{eql@equation*}%
6660 \eql@markline@amsthm@register{equation*}%
6661 }%
6662 \eql@provide@moveenv{equation*}{eql@equation*}%
6663 \eql@markline@amsthm@register{equation*}%
6664 \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
6665 \ifdefined\eql@tagging@on
6666 \AddToHook{begindocument/end}{%
6667 \eql@provide@moveenv{equation*}{eql@equation*}%
6668 \eql@markline@amsthm@register{equation*}%
6669 }%
6670 \fi
6671 \fi
6672 }

```

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

```

6673 \def\eql@provide@displaymath#1{%
6674 \eql@provide@onlyonce{displaymath}{#1}%
6675 \ifdefined\eql@tmp
6676 \eql@provide@moveenv{\eql@tmp}{eql@displaymath}%
6677 \eql@markline@amsthm@register{\eql@tmp}%
6678 \eql@tagging@register@luamml{\eql@tmp}%
6679 \else
6680 \eql@provide@moveenv{displaymath}{eql@displaymath}%
6681 \eql@markline@amsthm@register{displaymath}%
6682 \ifdefined\eql@tagging@on
6683 \AddToHook{begindocument/end}{%
6684 \eql@provide@moveenv{displaymath}{eql@displaymath}}%
6685 \fi
6686 \fi
6687 }

```

`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:

```

6688 \def\eql@provide@subequations#1{%
6689   \eql@provide@onlyonce{subequations}{#1}%
6690   \ifdefined\eql@tmp
6691     \eql@provide@moveenv{\eql@tmp}{eql@subequations}%
6692   \else
6693     \eql@amsmath@after{%
6694       \eql@provide@moveenv{subequations}{eql@subequations}%
6695     }%
6696     \AddToHook{package/hyperref/after}{%
6697       \AddToHook{cmd/subequations/before}[hyperref]{}%
6698       \AddToHook{cmd/subequations/after}[hyperref]{}%
6699       \RemoveFromHook{cmd/subequations/before}[hyperref]%
6700       \RemoveFromHook{cmd/subequations/after}[hyperref]%
6701       \AddToHook{begindocument/end}{%
6702         \eql@provide@moveenv{subequations}{eql@subequations}}%
6703     }%
6704     \eql@provide@moveenv{subequations}{eql@subequations}%
6705     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6706   \fi
6707 }

```

`\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:

```

6708 \def\eql@provide@sqr{%
6709   \let\[\eql@sqr@open
6710   \let\]\eql@sqr@close
6711   \eql@amsmath@after{%
6712     \let\[\eql@sqr@open
6713     \let\]\eql@sqr@close
6714   }%
6715   \ifdefined\eql@tagging@on
6716     \AddToHook{begindocument/end}{%
6717       \let\[\eql@sqr@open
6718       \let\]\eql@sqr@close
6719     }%
6720   \fi
6721 }

```

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

```

6722 \def\eql@provide@ang{%
6723   \let\<\eql@ang@open
6724   \let\>\eql@ang@close
6725 }

```

Interface.

`provide` (*key*) We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

6726 \eqld@define@key{provide}{equation}[]{\eqld@provide@equation{#1}}
6727 \eqld@define@key{provide}{equation*}[]{\eqld@provide@equationstar{#1}}
6728 \eqld@define@key{provide}{displaymath}[]{\eqld@provide@displaymath{#1}}
6729 \eqld@define@key{provide}{gather}[]{\eqld@provide@amsenv{gather}{#1}}
6730 \eqld@define@key{provide}{multline}[]{\eqld@provide@amsenv{multline}{#1}}
6731 \eqld@define@key{provide}{align}[]{\eqld@provide@amsenv{align}{#1}}
6732 \eqld@define@key{provide}{flalign}[]{\eqld@provide@amsenv{flalign}{#1}}
6733 \eqld@define@key{provide}{alignat}[]{\eqld@provide@amsenv{alignat}{#1}}
6734 \eqld@define@key{provide}{xalignat}[]{\eqld@provide@amsenv{xalignat}{#1}}
6735 \eqld@define@key{provide}{xxalignat}[]{\eqld@provide@amsenv{xxalignat}{#1}}
6736 \eqld@define@key{provide}{gather*}[]{\eqld@provide@amsenv{gather*}{#1}}
6737 \eqld@define@key{provide}{multline*}[]{\eqld@provide@amsenv{multline*}{#1}}
6738 \eqld@define@key{provide}{align*}[]{\eqld@provide@amsenv{align*}{#1}}
6739 \eqld@define@key{provide}{flalign*}[]{\eqld@provide@amsenv{flalign*}{#1}}
6740 \eqld@define@key{provide}{alignat*}[]{\eqld@provide@amsenv{alignat*}{#1}}
6741 \eqld@define@key{provide}{xalignat*}[]{\eqld@provide@amsenv{xalignat*}{#1}}
6742 \eqld@define@key{provide}{gathered}[]{\eqld@provide@amsbox{gathered}{#1}}
6743 \eqld@define@key{provide}{multlined}[]{\eqld@provide@multlined{#1}}
6744 \eqld@define@key{provide}{aligned}[]{\eqld@provide@amsbox{aligned}{#1}}
6745 \eqld@define@key{provide}{alignedat}[]{\eqld@provide@amsbox{alignedat}{#1}}
6746 \eqld@define@key{provide}{cases}[]{\eqld@provide@matrix{cases}\eqld@false{#1}}
6747 \eqld@define@key{provide}{matrix}[]{\eqld@provide@matrix{matrix}\eqld@false{#1}}
6748 \eqld@define@key{provide}{pmatrix}[]{\eqld@provide@matrix{pmatrix}\eqld@false{#1}}
6749 \eqld@define@key{provide}{bmatrix}[]{\eqld@provide@matrix{bmatrix}\eqld@true{#1}}
6750 \eqld@define@key{provide}{Bmatrix}[]{\eqld@provide@matrix{Bmatrix}\eqld@true{#1}}
6751 \eqld@define@key{provide}{vmatrix}[]{\eqld@provide@matrix{vmatrix}\eqld@true{#1}}
6752 \eqld@define@key{provide}{Vmatrix}[]{\eqld@provide@matrix{Vmatrix}\eqld@true{#1}}
6753 \eqld@define@key{provide}{subequations}[]{\eqld@provide@subequations{#1}}
6754 \eqld@define@key{provide}{sqr}[]{\eqld@provide@sqr}
6755 \eqld@define@key{provide}{ang}[]{\eqld@provide@ang}
6756 \eqld@define@key{provide}{eqref}[]{\eqld@provide@eqref{#1}}
6757 \eqld@define@key{provide}{tagform}[]{%
6758   \def\tagform##1{\maketag@@@{\eqld@tags@tagform{#1}}}%
6759 \eqld@define@key{provide}{maketag}[]{%
6760   \def\maketag@@@##1{\eqld@tags@taglayout{##1}}%

```

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

```

6761 \newcommand{\eqnlinesprovide}[1]{%
6762   \eqld@setkeys{provide}{#1}%
6763   \ignorespaces
6764 }

```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6765 \let\eqld@error@packageoption\gobble
```

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

```

6766 \DeclareOption{leqno}{\eqnlinesset{tagsleft}}
6767 \DeclareOption{fleqn}{\eqnlinesset{left}}

```

Pass undeclared options on to `keyval` processing:

```
6768 \DeclareOption*{\expandafter\eqnlinesset\expandafter{\CurrentOption}}
```


Set defaults for package:

```
6769 \eqldefaults@eqnlines
6770 \eqlequations@columns@set
6771 \eqlbox@columns@set
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the \TeX conditional parsing mechanism:

```
6772 \ifdefined\tagsleft@true\else
6773   \expandafter\newif\csname iftagsleft@\endcsname
6774 \fi
6775 \ifdefined\@fleqntrue\else
6776   \expandafter\newif\csname if@fleqn\endcsname
6777 \fi
```

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

```
6778 \eqlamsmath@after{%
6779   \ifnum\eqlprovide@opt@env=\tw@
6780     \iftagsleft@
6781       \eqnlineset{tags=left}%
6782     \else
6783       \eqnlineset{tags=right}%
6784     \fi
6785     \if@fleqn
6786       \eqnlineset{layout=left}%
6787     \else
6788       \eqnlineset{layout=center}%
6789     \fi
6790 \fi
6791 }
```

Process package options:

```
6792 \ProcessOptions
```

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

```
6793 \def\eqlerror@packageoption#1{%
6794   \eqlerror{may only use '#1' as a package option}%
6795 }
```

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

```
6796 \ifnum\eqlprovide@opt@env=\tw@
6797 \ifdefined\eqlprovide@opt@matrix
6798   \let\eqlprovide@opt@amsmathpatch\eqlfalse
6799 \fi\fi
6800 \ifdefined\eqlprovide@opt@backup
6801   \let\eqlprovide@opt@amsmathpatch\eqltrue
6802 \fi
6803 \ifdefined\eqlprovide@opt@amsmathpatch
6804   \eqlamsmath@fixends
6805   \eqlamsmath@fixmatrices
6806 \fi
```

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

```
6807 \ifdefined\eqlprovide@opt@backup\eqlprovide@backup\fi
```

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

```
6808 \ifnum\eq@provide@opt@env>\z@
6809   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6810 \fi
```

Provide `amsmath` equation environments if desired:

```
6811 \ifnum\eq@provide@opt@env=\tw@
6812   \eqnlinesprovide{%
6813     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6814     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6815     multlined,gathered,aligned,alignedat,%
6816     subequations}
6817 \fi
```

Provide symbolic shortcut `\<...\>` if desired:

```
6818 \ifdefined\eq@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
6819 \ifdefined\eq@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```

Provide `cases` and `matrix` environments if desired:

```
6820 \ifdefined\eq@provide@opt@matrix
6821   \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
6822 \fi
```