

The `eqnlines` Package

Source Code Documentation

Niklas Beisert

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

v1.1: 2026-06-07

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

Abstract

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

Contents

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

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

11.7 Options Processing	104
11.8 Environment	106
12 Single-Line Equation	107
12.1 Native Mode	107
12.2 Print	108
13 Multi-Line with Single Column	109
13.1 Measure	109
13.2 Column Placement	111
13.3 Print	111
14 Multi-Line with Multiple Columns	113
14.1 Support	113
14.2 Column Breaks	114
14.3 Transpose	115
14.4 Measure	117
14.5 Columns Placement	120
14.6 Print	120
15 Interface	123
15.1 Options Processing	123
15.2 Single-Line Main	126
15.3 Multi-Line Main	127
15.4 Equations Environment	129
16 Options	132
16.1 Selection Tools	132
16.2 Options Declarations	133
16.3 Parameter Presets	148
16.4 Component Selection	150
16.5 Global and Package Options	162

1 Implementation

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

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

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

2 General Support

In the following we describe general purpose supporting routines.

2.1 Debugging Messages

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

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

TODO: describe

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

2.2 Supporting Definitions

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

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

TODO: for comparison within `\ifx`

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

TODO: describe

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

2.3 Dollardollar Abstraction

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

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

```

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

```

2.4 Look-Ahead in Alignment

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

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

TODO: describe

```

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

```

`\eqL@spbgroup` The second challenge is addressed by enclosing the look-ahead in spurious groups¹ which
`\eqL@spgroup` protect against triggering ‘&’. The macros `\eqL@spbgroup` and `\eqL@spgroup` open and
`\eqL@srbgroup` close a spurious group. For some reason, the look-ahead mechanism requires further
`\eqL@sregroup` protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end.
 These adjustments are included in the macros `\eqL@srbgroup` and `\eqL@spgroup`:

```

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

```

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

```

99 \def\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@tablesaveinner{%
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@tablesave\@empty
224   \let\eql@tagging@tablesavealign\@empty
225   \let\eql@tagging@alignleft\@empty
226   \let\eql@tagging@aligncenter\@empty
227   \let\eql@tagging@alignright\@empty
228   \let\eql@tagging@start\@empty
229   \let\eql@tagging@end\@empty
230   \let\eql@tagging@register@luamml\@gobble
231   \let\eql@tagging@register@env\@gobble
232 \fi

```

2.8 Key-Value Processing

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

```

233 \RequirePackage{keyval}

```

Value Selection.

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

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

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

```

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

```

```

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

```

Decide between true and false or related pairs of values:

```

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

```

`\eq1@decide@if`

```

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

```

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

```

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

```

Key Declaration.

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

```

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

```

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

```

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

```

Options and Control Interface.

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

```
\eql@nextopt@apply
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 }
295 \def\eql@nextopt@apply#1{%
296   \eql@setkeys{#1}\eql@nextopt
297   \let\eql@nextopt\@empty
298 }
```

`\eqnaddopt`

```
299 \newcommand{\eqnaddopt}[1]{%
300   \ifx\eql@nextopt\@empty
301     \eql@append\eql@nextopt{#1}%
302   \else
303     \eql@append\eql@nextopt{, #1}%
304   \fi
305 }
```

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

```
306 \newcommand{\eqnlineset}[1]{%
307   \eql@setkeys{setup}{#1}%
308   \ignorespaces
309 }
```

`\eql@control@default`

```
310 \protected\def\eql@control@default{%
311   \eql@warn@here\eqncontrol
312   \@gobble
313 }
314 \let\eqncontrol\eql@control@default
```

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

```
315 \newcommand{\eql@control}[1]{%
316   \relax
317   \eql@setkeys{control}{#1}%
318   \ignorespaces
319 }
```

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.

```
320 \newtoks\eql@scan@reg@
```

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

```
321 \def\eql@scan@body@dump{\the\eql@scan@reg@}
322 \def\eql@scan@body@rescan{%
323   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
324 \let\eql@scan@body\eql@scan@body@dump
```

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

```
325 \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@`:

```
326 \def\eql@scan@env{%
327   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@env
328   \def\eql@scan@end{\expandafter\end\expandafter{\@currenvir}}}%
329   \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`:

```
330   \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
331   \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
332   \ifdefined\eql@scan@par
333     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
334   \else
335     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
336   \fi
337   \eql@scan@iterate
338 }
```

`\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:

```

339 \long\def\eql@scan@env@iterate#1\end#2{%
340   \edef\eql@scan@stack{%
341     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
342   \ifx\@empty\eql@scan@stack
343     \@checkend{#2}%
344     \eql@scan@addto{#1}%
345     \expandafter\let\eql@scan@iterate\eql@scan@env@org
346     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@env
347     \expandafter\eql@scan@call
348   \else
349     \eql@scan@addto{#1\end{#2}}%
350     \expandafter\eql@scan@iterate
351   \fi
352 }

```

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

```

353 \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.

```

354 \long\def\eql@scan@env@count#1\begin#2{%
355   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
356 }

```

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

```

357 \def\eql@scan@env@cancel{%
358   \@namedef{end\@currenvir}{\ignorespacesafterend}%
359   \eql@scan@end
360 }

```

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 `\]`:

```

361 \def\eql@scan@sqr{%
362   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@sqr
363   \def\eql@scan@end{\}%
364   \eql@scan@reg@{\}\def\eql@scan@stack{b}%
365   \let\eql@scan@sqr@org\[%\]
366   \ifdefined\eql@scan@par
367     \let\[\eql@scan@sqr@iterate%\]
368   \else
369     \let\[\eql@scan@sqr@iterate@nopar%\]
370   \fi
371   \[%\]
372 }

```

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

```

373 \long\def\eql@scan@sqr@iterate#1\]{%
374   \edef\eql@scan@stack{%
375     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%

```

```

376 \ifx\@empty\eql@scan@stack
377 \let\[\eql@scan@sqr@org%\]
378 \eql@scan@addto{#1}%
379 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@sqr
380 \expandafter\eql@scan@call
381 \else
382 \eql@scan@addto{#1\}}%
383 \expandafter\[%\]
384 \fi
385 }

```

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

```

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

```

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

```

387 \long\def\eql@scan@sqr@count#1\[#2{%\]
388 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
389 }

```

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

```

390 \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 `\>`:

```

391 \def\eql@scan@ang{%
392 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@ang
393 \def\eql@scan@end{\>}%
394 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
395 \let\eql@scan@ang@org\<%\>
396 \ifdefined\eql@scan@par
397 \let\<\eql@scan@ang@iterate%\>
398 \else
399 \let\<\eql@scan@ang@iterate@nopar%\>
400 \fi
401 \<%\>
402 }

```

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

```

403 \long\def\eql@scan@ang@iterate#1\>{%
404 \edef\eql@scan@stack{%
405 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
406 \ifx\@empty\eql@scan@stack
407 \let\<\eql@scan@ang@org%\>
408 \eql@scan@addto{#1}%
409 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@ang
410 \expandafter\eql@scan@call
411 \else
412 \eql@scan@addto{#1\>}%
413 \expandafter\<%\>
414 \fi
415 }

```

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

```
416 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}
```

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

```
417 \long\def\eql@scan@ang@count#1\<#2{\>
418   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
419 }
```

Interface. **TODO:** describe

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

```
420 \protected\def\eql@sqr@open{%
421   \ifmmode
422     \expandafter\eql@sqr@cancel
423   \else
424     \expandafter\eql@equations@sqr@open
425   \fi
426 }
```

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

```
427 \protected\def\eql@sqr@close{%
428   \eql@error{'\string\]' may only close '\string\[']'%\}
429 }
```

`\eql@sqr@cancel`

```
430 \def\eql@sqr@cancel{%
431   \eql@error@mathmode{\string\[...\string\]]}%
432   \let\eql@scan@call\eql@scan@sqrang@cancel
433   \eql@scan@sqr
434 }
```

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

```
435 \protected\def\eql@ang@open{%
436   \ifmmode
437     \expandafter\eql@box@ang@open
438   \else
439     \expandafter\eql@equations@ang@open
440   \fi
441 }
```

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

```
442 \protected\def\eql@ang@close{%
443   \eql@error{'\string\>' may only close '\string\<'}%\>
444 }
```

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:

```
445 \let\eql@tagsleft\eql@false
```

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

```
446 \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`:

```
447 \def\eql@layoutleftmargin{\leftmargini}
448 \def\eql@layoutleftmarginmax{.5\maxdimen}
449 \def\eql@layoutleftmarginmin{\z@}
```

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

```
450 \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

```
451 \newdimen\eql@tagmargin@
452 \let\eql@tagmargin@val\@undefined
453 \newdimen\eql@tagmargin@ratio@
454 \eql@tagmargin@ratio@\p@
455 \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:

```
456 \newdimen\eql@indent@
457 \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:

```
458 \newdimen\eql@paddingleft@
459 \newdimen\eql@paddingright@
460 \let\eql@paddingleft@val\@undefined
461 \let\eql@paddingright@val\@undefined
```

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

`\eqldisplay@marginleft`

`\eqldisplay@marginright`

```
462 \let\eqldisplay@linewidth\@undefined
463 \let\eqldisplay@marginleft\@undefined
464 \let\eqldisplay@marginright\@undefined
```

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

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

`\eql@box@condsep`

`\eql@box@matrixsep`

```
465 \def\eql@box@colsep{2em}
466 \def\eql@box@shortsep{1em}
467 \def\eql@box@condsep{\eql@box@shortsep}
468 \def\eql@box@matrixsep{\eql@box@shortsep}
```

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

`\eql@break@line@shortsep`

`\eql@break@col@sep`

`\eql@break@col@shortsep`

```
469 \def\eql@break@line@sep{2em minus 1em}
470 \def\eql@break@line@shortsep{1em}
471 \def\eql@break@col@sep{2em minus 1em}
472 \def\eql@break@col@shortsep{1em}
```

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

```
473 \let\eql@spread@reset\eql@false
474 \def\eql@spread@val{\jot}
475 \newdimen\eql@spread@
```

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

```
476 \newdimen\eql@tagfuzz@
477 \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:

```
478 \def\eqldisplay@height\@undefined
479 \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:

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

481 \def\eql@skip@mode@cont@above{2}
482 \def\eql@skip@mode@cont@below{0}

483 \def\eql@skip@mode@par@above{3}
484 \def\eql@skip@mode@par@below{0}

485 \def\eql@skip@mode@top@above{4}
486 \def\eql@skip@mode@top@below{0}

487 \newcount\eql@skip@mode@leave@
488 \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) 489 \newcount\eq@skip@mode@above@
\mode@below@ (counter) 490 \newcount\eq@skip@mode@below@
491 \let\eq@skip@force@above@\undefined
492 \let\eq@skip@force@below@\undefined
493 \let\eq@skip@custom@above@\undefined
494 \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`:

```

495 \def\eq@skip@long@above{\abovedisplayskip}
496 \def\eq@skip@long@below{\belowdisplayskip}
497 \def\eq@skip@short@above{\abovedisplayshortskip}
498 \def\eq@skip@short@below{\belowdisplayshortskip}
499 \def\eq@skip@cont@above{\eq@skip@short@above}
500 \def\eq@skip@cont@below{\eq@skip@short@below}
501 \def\eq@skip@par@above{\eq@skip@long@above}
502 \def\eq@skip@par@below{\eq@skip@long@below}
503 \def\eq@skip@top@above{\eq@skip@long@above}
504 \def\eq@skip@top@below{\eq@skip@long@below}
505 \def\eq@skip@med@above{\abovedisplayskip/2}
506 \def\eq@skip@med@below{\belowdisplayskip/2}
507 \def\eq@skip@tag@above{\z@skip}
508 \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:

```

509 \newdimen\eq@colsepmin@
510 \def\eq@colsepmin@val{1em}
511 \def\eq@colsepmax@val{.5\maxdimen}

```

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

```

512 \newdimen\eq@tagwidthmin@
513 \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.

```

514 \newdimen\eq@tagsepmin@
515 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

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

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

```
\eql@equations@ang@opt
  \eql@box@ang@opt
516 \def\eql@equations@sqr@opt{equation}
517 \def\eql@equations@ang@opt{columns}
518 \def\eql@box@ang@opt{columns}
```

Multi-Line Align Mode.

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

3.2 Registers

TODO: describe

General. **TODO:** describe

```
520 \newcount\eql@count@
521 \newdimen\eql@dimen@
522 \newskip\eql@skip@
```

TODO: describe

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

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

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

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

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

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

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

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

```
524 \newbox\eql@cellbox@
525 \newbox\eql@tagbox@
526 \newdimen\eql@cellwidth@
527 \newdimen\eql@prevwidth@
528 \newdimen\eql@tagwidth@
529 \newdimen\eql@prevdepth@
530 \newcount\eql@prevgraf@
```

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

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

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

```
531 \newdimen\eql@totalwidth@
532 \newdimen\eql@tagwidth@max@
533 \newdimen\eql@totalheight@
534 \newdimen\eql@topheight@
535 \newdimen\eql@bottomdepth@
```

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

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

```
536 \newdimen\eql@line@height@
537 \newdimen\eql@line@depth@
```

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

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

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

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

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

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

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

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

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

```
538 \newdimen\eql@line@width@
539 \newdimen\eql@line@avail@
540 \newdimen\eql@line@pos@
```

```

541 \newcount\eql@line@availsep@
542 \newcount\eql@line@widthsep@
543 \newcount\eql@line@possep@
544 \newdimen\eql@line@offset@
545 \newdimen\eql@line@prevdepth@
546 \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*)

```

547 \newcount\eql@row@
548 \newcount\eql@totalrows@
549 \newcount\eql@tagrows@

```

`\eql@column@`

`\eql@totalcolumns@`

```

550 \newcount\eql@column@
551 \newcount\eql@totalcolumns@

```

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

```

552 \newdimen\eql@colsep@

```

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

```

553 \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:

```

554 \newdimen\eql@display@firstavail@
555 \def\eql@display@firstavail@set#1{%
556   \ifnum\eql@row@=\@ne
557     \global\eql@appendexpand\eql@display@container{%
558       \eql@display@firstavail@\the#1\relax}%
559   \fi
560 }

```

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

```

561 \newdimen\eql@display@aboveextend@
562 \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:

```

563 \ifdefined\measuring@true\else
564   \expandafter\newif\csname ifmeasuring@\endcsname
565 \fi
566 \AddToHook{package/hyperref/after}{%
567   \ifdefined\Hy@ifnotmeasuring
568     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}%
569   \fi
570 }
```

`\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:

```

571 \ifdefined\@displaytrue\else
572   \expandafter\newif\csname if@display\endcsname
573   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
574 \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

```

575 \let\eql@hook@blockbefore\@empty
576 \let\eql@hook@blockafter\@empty
577 \let\eql@hook@blockin\@empty
578 \let\eql@hook@blockout\@empty
579 \let\eql@hook@linein\@empty
580 \let\eql@hook@lineout\@empty
581 \let\eql@hook@colin\@empty
582 \let\eql@hook@colout\@empty
583 \let\eql@hook@eqin\@empty
584 \let\eql@hook@eqout\@empty
585 \let\eql@hook@innerleft\@empty
586 \let\eql@hook@innerright\@empty
587 \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` 588 \let\eql@punct@col\@empty

`\eql@punct@cases`

```

589 \let\eql@punct@line\@undefined
590 \let\eql@punct@block\@undefined
591 \let\eql@punct@next\@undefined
592 \let\eql@punct@top\@undefined
593 \let\eql@punct@cases\@empty

```

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

```

594 \let\eql@punct@class\@empty
595 \let\eql@punct@sep\@empty

```

`\eql@punct@top@set` **TODO:** describe

```

\eql@punct@top@stop
\eql@punct@top@reset
596 \def\eql@punct@top@set{\let\eql@punct@top\eql@punct@block}
597 \def\eql@punct@top@stop{\let\eql@punct@top\relax}
598 \def\eql@punct@top@reset{\let\eql@punct@top\@undefined}

```

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

```

599 \def\eql@punct@tilde{~}
600 \def\eql@punct@set#1#2{%
601   \def#1{#2}%
602   \ifx#1\eql@relax
603     \let#1\@undefined
604   \fi
605   \ifx#1\eql@punct@tilde
606     \let#1\@empty
607   \fi
608 }
609 \def\eql@punct@clear{%
610   \let\eql@punct@col\@empty
611   \let\eql@punct@line\@empty
612   \let\eql@punct@block\@empty
613 }
614 \def\eql@punct@next@clear{\let\eql@punct@next\@empty}

```

Set the punction 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

```

615 \def\eql@punct@adopt{%
616   \eql@ifstar@tight\eql@punct@adopt@relax\eql@punct@adopt@set}
617 \def\eql@punct@adopt@set#1{\eqnadopt{punct={#1}}\ignorespaces}
618 \def\eql@punct@adopt@relax{\eqnadopt{punct*}\ignorespaces}

```

TODO: describe

```

619 \def\eql@punct@setnext{%
620   \eql@ifstar@tight\eql@punct@setnext@relax\eql@punct@setnext@set}
621 \def\eql@punct@setnext@set#1{%
622   \eql@punct@set\eql@punct@next{#1}%
623   \ifdefined\eql@punct@next\else\let\eql@punct@next\relax\fi
624   \ignorespaces}
625 \def\eql@punct@setnext@relax{\let\eql@punct@next\relax\ignorespaces}

```

`\eqnpunct` **TODO:** describe

```

626 \let\eqnpunct\eql@punct@adopt

```

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

```
627 \def\eql@punct@fill@next#1{%
628   \ifdefined\eql@punct@next
629     \ifx\eql@punct@next\relax
630       \let\eql@punct@next\@undefined
631     \fi
632   \else
633     \ifx\eql@punct@top\relax\else
634       \let\eql@punct@next#1%
635     \fi
636   \fi
637 }
```

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

```
638 \def\eql@punct@output@next{%
639   \ifx\eql@punct@next\@empty\else
640     \ifmmode\eql@punct@class\fi
641     \eql@punct@sep
642     \eql@punct@next
643   \fi
644   \let\eql@punct@next\@undefined
645 }
```

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

```
646 \def\eql@punct@print@next{%
647   \ifdefined\eql@punct@next
648     \eql@punct@output@next
649   \fi
650 }
```

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

```
651 \def\eql@punct@apply@next{%
652   \ifdefined\eql@punct@next
653     \eql@punct@output@next
654     \eql@punct@top@stop
655   \fi
656 }
```

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

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

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

```
661 \def\eql@punct@apply@col{%
662   \eql@punct@fill@next\eql@punct@col
663   \eql@punct@apply@next
664 }
```

Output the punctuation for the present line unless disabled:

`\eql@punct@print@line`

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

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

`\eql@punct@apply@line`

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

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

```
673 \def\eql@punct@apply@block{%
674   \eql@punct@fill@next\eql@punct@block
675   \eql@punct@apply@next
676 }
```

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

```
677 \def\eql@punct@apply@top{%
678   \eql@punct@fill@next\eql@punct@top
679   \eql@punct@print@next
680   \global\eql@punct@top@stop
681 }
```

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

```
682 \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:

```
683 \def\eql@class@innerright@sel@{%
684   \ifdim\eql@prevwidth@=\z@
685     \eql@class@innerlead
686   \else
687     \eql@class@innerright
688   \fi
689 }
```

`\@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`

```
690 \def\eql@class@innerleft@set#1{%
691   \def\eql@class@innerleft{#1}%
692 }
```

```

693 \def\eq@class@innerright@set#1{%
694   \def\eq@class@innerright{#1}%
695   \let\eq@class@innerright@sel\eq@class@innerright
696 }
697 \def\eq@class@innerlead@set#1{%
698   \def\eq@class@innerlead{#1}%
699   \let\eq@class@innerright@sel\eq@class@innerright@sel@
700 }

701 \def\eq@class@rel@symb{=}
702 \def\eq@class@rel@break#1{\eq@class@rel@start{#1}\mathclose{}}
703 \def\eq@class@rel@start#1{\mathrel{\phantom{#1}}}
704 \def\eq@class@rel@break@set#1{\def\eq@class@rel@break##1{#1}}
705 \def\eq@class@rel@start@set#1{\def\eq@class@rel@start##1{#1}}
706 \def\eq@class@rel@break@skip{\thickmuskip-\medmuskip}
707 \def\eq@class@rel@amp{&}
708 \def\eq@class@rel@amprelax{&\relax}
709 \def\eq@class@rel@relaxamp{\relax&}
710 \def\eq@class@rel@ordamp{{}&}
711 \def\eq@class@rel@make#1{%
712   \ifdefined\eq@class@rel@after
713     \def\eq@tmp{&#1}%
714     \ifx\eq@tmp\eq@class@rel@amp
715       \def\eq@tmp{&\eq@class@rel@break\eq@class@rel@symb
716         \mskip\muexpr\eq@class@rel@break@skip\relax}%
717     \else\ifx\eq@tmp\eq@class@rel@amprelax
718       \def\eq@tmp{&\eq@class@rel@start\eq@class@rel@symb}%
719     \fi\fi
720   \else
721     \def\eq@tmp{#1&}%
722     \ifx\eq@tmp\eq@class@rel@amp
723       \def\eq@tmp{&\mathclose{}}%
724       \mskip\muexpr\eq@class@rel@break@skip\relax}%
725     \fi
726     \ifx\eq@tmp\eq@class@rel@relaxamp
727       \def\eq@tmp{&\mathclose{}\mathopen{}\mathrel{}}%
728     \fi
729     \ifx\eq@tmp\eq@class@rel@ordamp
730       \def\eq@tmp{&\mathclose{}}}%
731     \fi
732   \fi
733   \eq@tmp
734 }

```

`\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`):

```

735 \def\eq@class@ampeq{%
736   \eq@class@innerleft@set{}%
737   \eq@class@innerright@set{}}%
738   \let\eq@class@rel@after\eq@true
739 }
740 \def\eq@class@eqamp{%
741   \eq@class@innerleft@set{}%
742   \eq@class@innerright@set{\mathrel{}}%
743   \eq@class@innerlead@set{\mathclose{}}%
744   \let\eq@class@rel@after\eq@false
745 }
746 \eq@class@ampeq

```

4.3 Framed Cells

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

```
747 \let\eql@frame@cmd\@undefined
748 \newdimen\eql@frame@margin@
749 \def\eql@frame@set[#1]{%
750   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
751   \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbx}
752   \def\eql@frame@measure{%
753     \setbox\z@\hbox{\eql@frame@cmd{}}%
754     \eql@frame@margin@.5\wd\z@
755   }
756   \def\eql@frame@print{%
757     \setbox\eql@cellbox@\hbox{%
758       \eql@frame@cmd{\unhbox\eql@cellbox@}%
759     }%
760   }
761   \def\eql@frame@adjust{%
762     \setbox\eql@cellbox@\hbox{%
763       \eql@frame@cmd{%
764         \unhbox\eql@cellbox@
765         \unkern
766         \unskip
767       }%
768       \hfil
769       \kern\z@
770     }%
771   }
```

4.4 Single-Line Composition

TODO: describe

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

```

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

```

TODO: describe

```

806 \def\eql@break@test@setopt{\let\eql@break@test\eql@break@testopt}
807 \def\eql@break@test@setall{\let\eql@break@test\eql@break@testall}
808 \def\eql@break@cr@test@setopt{\let\eql@break@cr@test\eql@break@testopt}
809 \def\eql@break@cr@test@setall{\let\eql@break@cr@test\eql@break@testall}
810 \def\eql@break@amp@test@setopt{\let\eql@break@amp@test\eql@break@testopt}
811 \def\eql@break@amp@test@setall{\let\eql@break@amp@test\eql@break@testall}

```

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

```

812 \def\eql@break@testopt#1{\eql@ifstar@tight
813 {\let\eql@break@sep\eql@break@shortsep#1}{\eql@break@testopt@arg{#1}}}
814 \def\eql@break@testopt@arg#1{\eql@ifnextchar@tight[%
815 {\eql@break@testopt@set{#1}}{#1}}
816 \def\eql@break@testopt@set#1[#2]{\def\eql@break@sep{#2}#1}

```

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

@break@testall@parse

```

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

```

```

844 \let\eql@parseopt@next\eql@parseopt@rel symb
845 \fi
846 \ifx\eql@parseopt@token;%
847 \let\eql@parseopt@next\eql@parseopt@rel cont
848 \fi
849 \ifx\eql@parseopt@token:%
850 \let\eql@parseopt@next\eql@parseopt@rel start
851 \fi
852 \ifx\eql@parseopt@token|%
853 \let\eql@parseopt@next\eql@parseopt@rel ord
854 \fi
855 \ifx\eql@parseopt@token?%
856 \let\eql@parseopt@next\eql@break@parse@rel
857 \fi
858 }
859 \def\eql@break@parse@val[#1]{%
860 \def\eql@break@sep{#1}\eql@parseopt@peek}
861 \def\eql@break@parse@star#1{%
862 \let\eql@break@sep\eql@break@shortsep\eql@parseopt@peek}
863 \def\eql@break@parse@rel#1#2{%
864 \def\eql@class@rel@composed{#2}\eql@parseopt@end}

```

`\eql@break@process`

```

865 \def\eql@break@process{%
866 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
867 \ifdefined\eql@class@rel@composed
868 \eql@class@rel@composed
869 \else
870 \eql@break@print
871 \hspace{\glueexpr\eql@break@sep\relax}%
872 \fi
873 }

```

`\eql@break@join`

```

874 \def\eql@break@join{\eql@srbgroup
875 \eql@ifstar@tight
876 {\eql@break@join@opt[\eql@break@col@shortsep]}%
877 {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
878 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
879 \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

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

`\eqnbreak`

`\eqnjoin`

```

880 \newcommand{\eqnsep}{\eql@break@col}
881 \newcommand{\eqnbreak}{\eql@break@line}
882 \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

```

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

```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```
884 \let\eq\@tags@autolabel\eq@false
885 \let\eq\@tags@autotag\eq@true
886 \let\eq\@tags@warn\eq@true

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

888 \let\eq\@tagpos@doconvert\eq@false

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

Registers.

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

891 \let\eq\@numbering@active\eq@false
892 \let\eq\@numbering@multi\eq@true

893 \let\eq\@tags@container\@undefined
894 \def\eq\@tags@container@clear{%
895   \let\eq\@tags@label\@undefined
896   \let\eq\@tags@name\@undefined
897   \let\eq\@tags@tag\@undefined
898   \let\eq\@tags@ref\@undefined
899   \let\eq\@tags@anchor\@empty
900   \eq\@tagpos@shift@\z@
901   \eq\@tagpos@smashup@\z@
902   \eq\@tagpos@smashdown@\z@
903   \let\eq\@tagpos@reserve\eq@true
904 }

905 \let\eq\@tags@label\@undefined
906 \let\eq\@tags@name\@undefined
907 \let\eq\@tags@tag\@undefined
908 \let\eq\@tags@ref\@undefined
909 \let\eq\@tags@frame@cmd\@firstofone

\eq\@tags@glabel@ (counter)

910 \newcount\eq\@tags@glabel@
911 \eq\@tags@glabel@\z@
912 \def\eq\@tags@glabel{equation.\eq-\the\eq\@tags@glabel@}
913 \def\eq\@tags@glabel@step{\global\advance\eq\@tags@glabel@\@ne}

914 \let\eq\@tagpos@continuous\eq@false

915 \newcount\eq\@tagpos@row@
916 \newcount\eq\@tagpos@prevrow@
917 \newdimen\eq\@tagpos@shift@
918 \newdimen\eq\@tagpos@smashup@
919 \newdimen\eq\@tagpos@smashdown@
```

```

920 \newdimen\eql@tagpos@current@
921 \newdimen\eql@tagpos@plain@
922 \newdimen\eql@tagpos@raised@
923 \newdimen\eql@tagpos@target@
924 \newdimen\eql@tagpos@headroom@
925 \newdimen\eql@tagpos@footroom@

```

5.2 Schemes

TODO: describe

Table.

```

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

941 \let\eql@numbering@mode\eql@numbering@tab@all
942 \let\eql@numbering@mode@multi\eql@numbering@tab@all
943 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```

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

```

TODO: describe

```

957 \let\eql@numbering@tab@a\eql@numbering@tab@all
958 \let\eql@numbering@tab@s\eql@numbering@tab@sub
959 \let\eql@numbering@tab@f\eql@numbering@tab@first
960 \let\eql@numbering@tab@l\eql@numbering@tab@last
961 \let\eql@numbering@tab@o\eql@numbering@tab@out
962 \let\eql@numbering@tab@i\eql@numbering@tab@in
963 \let\eql@numbering@tab@h\eql@numbering@tab@here

```

```

964 \let\eql@numbering@tab@t\eql@numbering@tab@top
965 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
966 \let\eql@numbering@tab@c\eql@numbering@tab@center
967 \let\eql@numbering@tab@m\eql@numbering@tab@median
968 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
969 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
970 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
971 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

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

```

TODO: describe

```

973 \def\eql@numbering@init@sub{%
974   \let\eql@numbering@multi\eql@true
975   \ifdefined\eql@subequations@active
976     \let\eql@numbering@mode\eql@numbering@tab@all
977   \else
978     \let\eql@numbering@subeq@use\eql@true
979   \fi
980 }

981 \def\eql@numbering@init@first{\eql@tagpos@row@{\@ne}
982 \def\eql@numbering@init@last{\eql@tagpos@row@{\@MM}
983 \def\eql@numbering@init@here{\eql@tagpos@row@{\m@ne}

```

TODO: describe

```

984 \def\eql@numbering@init@in{%
985   \ifdefined\eql@tagsleft
986     \eql@numbering@init@last
987   \else
988     \eql@numbering@init@first
989   \fi
990 }

```

TODO: describe

```

991 \def\eql@numbering@init@out{%
992   \ifdefined\eql@tagsleft
993     \eql@numbering@init@first
994   \else
995     \eql@numbering@init@last
996   \fi
997 }

```

TODO: describe

```

998 \def\eql@tagpos@eval@middle{%
999   \ifnum\eql@tagpos@row@=\z@
1000     \eql@tagpos@row@\numexpr(\eql@totalrows@
1001       +\ifdefined\eql@tagsleft\z@\else\@ne\fi)/\tw@\relax
1002   \fi
1003 }

```

TODO: describe

```

1004 \def\eql@tagpos@eval@best{%
1005   \ifnum\eql@tagpos@row@=\z@

```



```

1006 \let\eq@numbering@best@use\eq@true
1007 \eq@numbering@init@out
1008 \fi
1009 }

```

TODO: describe

```

1010 \def\eq@numbering@init@continuous{\let\eq@tagpos@continuous\eq@true}

```

TODO: describe

```

1011 \let\eq@numbering@init@top\eq@numbering@init@continuous
1012 \def\eq@tagpos@eval@top{%
1013 \eq@tagpos@current@z@
1014 }

```

TODO: describe

```

1015 \let\eq@numbering@init@bottom\eq@numbering@init@continuous
1016 \def\eq@tagpos@eval@bottom{%
1017 \eq@tagpos@current@\dimexpr\eq@totalheight@
1018 -\eq@tagheight@block@-\eq@tagdepth@block@\relax
1019 }

```

TODO: describe

```

1020 \let\eq@numbering@init@center\eq@numbering@init@continuous
1021 \def\eq@tagpos@eval@center{%
1022 \ifnum\eq@totalrows@=\@ne
1023 \eq@tagpos@row@\@ne
1024 \fi
1025 \eq@tagpos@current@\dimexpr(\eq@totalheight@
1026 -\eq@tagheight@block@-\eq@tagdepth@block@)/\tw@\relax
1027 }

```

TODO: describe

```

1028 \let\eq@numbering@init@centerone\eq@numbering@init@continuous
1029 \def\eq@tagpos@eval@centerone{%
1030 \eq@tagpos@current@\dimexpr(\eq@totalheight@
1031 -\eq@tagheight@block@-\eq@tagdepth@block@)/\tw@\relax
1032 }

```

TODO: describe

```

1033 \let\eq@numbering@init@baseline\eq@numbering@init@continuous
1034 \def\eq@tagpos@eval@baseline{%
1035 \eq@tagpos@current@\dimexpr(\eq@totalheight@
1036 +\eq@topheight@-\eq@bottomdepth@)/\tw@-\eq@tagheight@block@\relax
1037 }

```

TODO: describe

```

1038 \let\eq@numbering@init@median\eq@numbering@init@continuous
1039 \def\eq@tagpos@eval@median{%
1040 \ifnum\eq@tagpos@row@=z@
1041 \ifodd\eq@totalrows@
1042 \eq@tagpos@row@\numexpr(\eq@totalrows@+\@ne)/\tw@\relax
1043 \else
1044 \eq@tagpos@row@\numexpr(\eq@totalrows@+\tw@)/\tw@\relax
1045 \eq@dimensions@get\eq@tagpos@row@
1046 \advance\eq@tagpos@shift@\dimexpr\eq@line@height@
1047 +(\eq@line@interline@-\eq@tagheight@block@

```

```

1048      +\eql@tagdepth@block@)/\tw@/relax
1049    \fi
1050    \ifnum\eql@totalrows@=\@ne
1051      \eql@tagpos@row@\@ne
1052    \else
1053      \eql@tagpos@adjust@eval@convert
1054      \eql@tagpos@row@\z@
1055    \fi
1056  \fi
1057 }

```

Selection.

```

1058 \def\eql@numbering@set#1{%
1059   \ifcsname eql@numbering@tab@#1\endcsname
1060     \expandafter\let\expandafter\eql@numbering@mode
1061     \csname eql@numbering@tab@#1\endcsname
1062   \ifx\eql@numbering@mode\eql@numbering@tab@all
1063     \let\eql@numbering@mode@multi\eql@numbering@mode
1064   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
1065     \let\eql@numbering@mode@multi\eql@numbering@mode
1066   \else
1067     \let\eql@numbering@mode@single\eql@numbering@mode
1068   \fi\fi
1069 \else
1070   \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
1071   \let\eql@numbering@mode\eql@numbering@tab@all
1072 \fi
1073 }

```

TODO: describe

```

1074 \def\eql@numbering@init{%
1075   \let\eql@numbering@multi\eql@false
1076   \let\eql@tagpos@continuous\eql@false
1077   \let\eql@numbering@subeq@use\eql@false
1078   \let\eql@numbering@best@use\eql@false
1079   \eql@tagpos@row@\z@
1080   \csname eql@numbering@init@\eql@numbering@mode\endcsname
1081   \ifdefined\eql@numbering@active
1082     \let\eql@numbering@eqnswinit@eqnswtrue
1083   \else
1084     \let\eql@numbering@eqnswinit@eqnswfalse
1085   \fi
1086   \let\eql@numbering@active\eql@false
1087 }

```

5.3 Interface

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

```

1088 \eql@amsmath@after{%
1089   \let\eql@print@eqnum@default\print@eqnum
1090   \let\eql@incr@eqnum@default\incr@eqnum
1091 }

```

TODO: describe

```

1092 \protected\def\donumber{%
1093   \if@eqnsw\else
1094     \global\@eqnswtrue
1095     \ifx\print@eqn\@empty
1096       \global\let\print@eqn\eq\print@eqnum@default
1097     \fi
1098     \ifx\incr@eqn\@empty
1099       \global\let\incr@eqn\eq\incr@eqnum@default
1100     \fi
1101   \fi
1102 }

```

TODO: reconsider operation

\numberhere

```

1103 \protected\def\eq\@numberhere{%
1104   \ifdefined\eq\@numbering@multi
1105     \global\@eqnswtrue
1106   \else
1107     \global\eq\@tagpos@row@\eq\@row@
1108   \fi
1109 }

```

TODO: describe

\numbernext

```

1110 \protected\def\eq\@numbernext{%
1111   \ifdefined\eq\@numbering@multi
1112     \global\@eqnswfalse
1113   \else
1114     \ifdefined\eq\@tagpos@continuous\else
1115       \ifnum\eq\@tagpos@row@=\eq\@row@
1116         \global\advance\eq\@tagpos@row@\@ne
1117       \fi
1118     \fi
1119   \fi
1120 }

```

Activation Trigger.

```

1121 \def\eq\@tags@autoenable{%
1122   \global\@eqnswtrue
1123   \ifnum\eq\@tagpos@row@=\m@ne
1124     \numberhere
1125   \fi
1126 }

```

Labels. **TODO:** describe

\eq\@label@org

```

1127 \let\eq\@label@org\label

```

TODO: describe

```

1128 \def\eq\@label@gobble{\eq\@ampprotect\eq\@testopt@tight\eq\@gobbleoptone{}}

```

TODO: describe

```
1129 \protected\def\eql@label{%
1130   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
1131 }
```

TODO: describe

```
1132 \def\eql@tags@add@labelname[#1]#2{%
1133   \def\eql@tmp{#1}%
1134   \ifx\eql@tmp\eql@testopt@default\else
1135     \eql@tags@add@name{#1}%
1136   \fi
1137   \eql@tags@add@label{#2}%
1138 }
```

TODO: describe

```
1139 \def\eql@tags@set@label#1{%
1140   \ifdefined\eql@tags@warn
1141     \ifdefined\eql@tags@label
1142       \eql@warn@label@multiple{#1}%
1143     \fi
1144   \fi
1145   \def\eql@tags@label{#1}%
1146 }
```

TODO: describe

```
1147 \def\eql@tags@set@name#1{%
1148   \ifdefined\eql@tags@warn
1149     \ifdefined\eql@tags@name
1150       \eql@warn@name@multiple
1151     \fi
1152   \fi
1153   \def\eql@tags@name{#1}%
1154 }
```

TODO: describe

```
1155 \def\eql@tags@add@label#1{%
1156   \ifdefined\eql@tags@autolabel
1157     \eql@tags@autoenable
1158   \fi
1159   \global\eql@appendexpand\eql@tags@container{%
1160     \noexpand\eql@tags@set@label{#1}}%
1161 }
```

TODO: describe

```
1162 \def\eql@tags@add@name#1{%
1163   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1164   \global\eql@appendmacro\eql@tags@container\eql@tmp
1165 }
```

TODO: describe

```
1166 \def\eql@tags@add@block@label#1{%
1167   \eql@appendexpand\eql@tags@container@block{%
1168     \noexpand\eql@tags@set@label{#1}}%
1169 }
```

TODO: describe

```

1170 \def\eql@tags@addblock@name#1{%
1171   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1172   \eql@appendmacro\eql@tags@container@block\eql@tmp
1173 }

```

Tags. **TODO:** describe

`\eql@tag@default`

```

1174 \protected\def\eql@tag@default{%
1175   \eql@warn@here\tag
1176   \eql@tag@gobble
1177 }
1178 \let\tag\eql@tag@default

```

`\eql@tag@gobble`

```

1179 \def\eql@tag@gobble{%
1180   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}

```

TODO: describe

```

1181 \protected\def\eql@tag{%
1182   \eql@ampprotecttwo\eql@teststaropt@tight
1183   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}%
1184   \eql@testopt@default
1185 }

```

`\eql@tags@add@tagref`

```

1186 \def\eql@tags@add@tagref[#1]#2{%
1187   \def\eql@tmp{#1}%
1188   \ifx\eql@tmp\eql@testopt@default\else
1189     \eql@tags@add@ref{#1}%
1190   \fi
1191   \eql@tags@add@tag{#2}%
1192 }

```

TODO: describe

```

1193 \def\eql@tags@set@tag#1{%
1194   \ifdefined\eql@tags@warn
1195     \ifdefined\eql@tags@tag
1196       \eql@warn@tag@multiple
1197     \fi
1198   \fi
1199   \def\eql@tags@tag{#1}%
1200 }

```

TODO: describe

```

1201 \def\eql@tags@set@ref#1{%
1202   \ifdefined\eql@tags@warn
1203     \ifdefined\eql@tags@ref
1204       \eql@warn@ref@multiple
1205     \fi
1206   \fi
1207   \def\eql@tags@ref{#1}%
1208 }

```

TODO: describe

```
1209 \def\eql@tags@add@tag#1{%
1210   \ifdefined\eql@tags@autotag
1211     \eql@tags@autoenable
1212   \fi
1213   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
1214   \global\eql@appendmacro\eql@tags@container\eql@tmp
1215 }
```

TODO: describe

```
1216 \def\eql@tags@add@ref#1{%
1217   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
1218   \global\eql@appendmacro\eql@tags@container\eql@tmp
1219 }
```

tags@add@tagform@off

```
1220 \def\eql@tags@add@tagform@off{%
1221   \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
1222 }
```

TODO: describe

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

TODO: describe

```
1227 \def\eql@tags@addblock@ref#1{%
1228   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
1229   \eql@appendmacro\eql@tags@container@block\eql@tmp
1230 }
```

TODO: describe

```
1231 \def\eql@tags@addblock@tagform@off{%
1232   \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
1233 }
```

Raise Tags.

\raisetag

```
1234 \def\eql@raisetag@default{%
1235   \eql@warn@here\raisetag
1236   \eql@raisetag@gobble
1237 }

1238 \def\eql@raisetag@gobble{%
1239   \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble
1240 }
```

TODO: describe

```
1241 \eql@amsmath@let\raisetag\eql@raisetag@default

1242 \def\eql@raisetag{%
1243   \eql@ampprotecttwo\eql@ifstar@tight\eql@tags@add@raiseshift\eql@raisetag@test
1244 }
```

```

1245 \def\eq@raisetag@test#1{%
1246   \def\eq@tmpa{#1}\def\eq@tmpb{!}%
1247   \ifx\eq@tmpa\eq@tmpb
1248     \eq@tags@add@forceraise
1249   \else
1250     \eq@tags@add@raisesmash{#1}%
1251   \fi
1252 }

1253 \def\eq@tags@add@raiseshift#1{%
1254   \global\eq@appendexpand\eq@tags@container{%
1255     \advance\eq@tagpos@shift@the\glueexpr#1\relax\relax}%
1256 }

1257 \def\eq@tags@add@raisesmash#1{%
1258   \dimen@glueexpr#1\relax
1259   \ifdim\dimen@<\z@
1260     \global\eq@appendexpand\eq@tags@container{%
1261       \advance\eq@tagpos@smashdown@the\dimen@\relax}%
1262   \else
1263     \global\eq@appendexpand\eq@tags@container{%
1264       \advance\eq@tagpos@smashup@the\dimen@\relax}%
1265   \fi
1266 }

1267 \def\eq@tags@add@forceraise{%
1268   \global\eq@append\eq@tags@container{\let\eq@tagpos@reserve\eq@false}%
1269 }

```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```

1270 \def\eq@numbering@settools{%
1271   \let\label\eq@label
1272   \let>tag\eq@tag
1273   \let\raisetag\eq@raisetag
1274   \let\numberhere\eq@numberhere
1275   \let\numbernext\eq@numbernext
1276 }

```

TODO: not necessary anymore

```

1277 \def\eq@numbering@settools@gobble{%
1278   \let\label\eq@label@gobble
1279   \let>tag\eq@tag@gobble
1280   \let\raisetag\eq@raisetag@gobble
1281   \let\numberhere\relax
1282   \let\numbernext\relax
1283 }

1284 \def\eq@numbering@autoblock{%
1285   \begingroup
1286     \let\eq@tags@warn\eq@false
1287     \eq@tags@container@block
1288     \ifdefined\eq@tags@autolabel

```

```

1289     \ifdefined\eql@tags@label
1290     \global\@eqnswtrue
1291     \fi
1292   \fi
1293   \ifdefined\eql@tags@autotag
1294     \ifdefined\eql@tags@tag
1295       \global\@eqnswtrue
1296     \fi
1297   \fi
1298 \endgroup
1299 }

1300 \def\eql@numbering@warnunused{%
1301   \ifdefined\eql@tags@label
1302     \eql@warn@label@unused
1303   \fi
1304   \ifdefined\eql@tags@name
1305     \eql@warn@name@unused
1306   \fi
1307   \ifdefined\eql@tags@tag
1308     \eql@warn@tag@unused
1309   \fi
1310   \ifdefined\eql@tags@erf
1311     \eql@warn@ref@unused
1312   \fi
1313 }

```

Single Line. **TODO:** describe

```

1314 \def\eql@numbering@single@init{%
1315   \let\eql@numbering@multi\eql@false
1316   \eql@numbering@settools
1317   \eql@numbering@eqnswinit
1318   \eql@numbering@autoblock
1319   \global\let\eql@tags@container\eql@tags@container@block
1320   \let\eql@tags@warn\eql@true
1321 }

1322 \def\eql@numbering@single@eval{%
1323   \ifnum\eql@tagpos@row@=\m@ne
1324     \@eqnswfalse
1325   \fi
1326 }

```

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

```

1327 \def\eql@numbering@measure@init{%
1328   \eql@numbering@settools
1329   \ifdefined\eql@numbering@multi\else
1330     \eql@numbering@eqnswinit
1331     \eql@numbering@autoblock
1332   \fi
1333   \global\let\eql@tags@container\eql@tags@container@block
1334   \let\eql@tags@warn\eql@true
1335 }

```

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


```

1336 \def\eql@numbering@measure@line@begin{%
1337   \ifdefined\eql@numbering@multi
1338     \global\eql@numbering@eqnswinit
1339   \fi
1340 }

```

TODO: describe

```

1341 \def\eql@numbering@measure@blocktag{%
1342   \ifdefined\eql@numbering@multi
1343     \eqnswfalse
1344   \else
1345     \ifnum\eql@tagpos@row@=\m@ne
1346       \eqnswfalse
1347     \fi
1348     \ifnum\eql@totalrows@=\z@
1349       \eqnswfalse
1350     \fi
1351   \fi
1352 }

```

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!

```

1353 \def\eql@numbering@print@init{%
1354   \let\eql@tags@warn\eql@false
1355   \ifdefined\eql@numbering@multi
1356     \eql@numbering@settools
1357     \global\let\eql@tags@container\eql@tags@container@block
1358   \else
1359     \let\eql@tags@container@block\eql@tags@container
1360     \eql@numbering@settools@gobble
1361   \fi
1362 }

```

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

```

1363 \def\eql@numbering@print@block@begin{%
1364   \ifdefined\eql@numbering@multi\else
1365     \ifnum\eql@tagpos@row@>\z@
1366       \eql@tags@makeblockanchor
1367       \global\eql@appendexpand\eql@tags@container@block{%
1368         \def\noexpand\eql@tags@anchor{%
1369           \unexpanded\expandafter{\eql@tags@anchor}}}%
1370     \fi
1371   \fi
1372   \ifdefined\eql@numbering@subeq@use
1373     \eql@tags@printsbeqlabel
1374   \fi
1375 }

```

TODO: describe

```

1376 \def\eql@numbering@print@line@begin{%
1377   \ifdefined\eql@numbering@multi
1378     \global\eql@numbering@eqnswinit
1379   \fi
1380 }

```

TODO: describe

```
1381 \def\eql@numbering@print@line@eval{%
1382   \ifdefined\eql@numbering@multi
1383     \if@eqnsw
1384       \eql@tags@container
1385     \fi
1386   \else
1387     \ifnum\eql@tagpos@row@=\eql@row@
1388       \@eqnswtrue
1389       \eql@tags@container@block
1390     \else
1391       \@eqnswfalse
1392     \fi
1393   \fi
1394 }
```

5.5 Positioning

TODO: describe

```
1395 \def\eql@tagpos@single@eval{%
1396   \if@eqnsw
1397     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1398     \ifnum\eql@tagpos@row@>\@ne
1399       \eql@tagpos@row@\@ne
1400     \fi
1401     \ifdefined\eql@tagpos@doconvert
1402       \let\eql@tagpos@continuous\eql@true
1403     \fi
1404     \ifdefined\eql@tagpos@continuous
1405       \eql@tagpos@single@eval@continuous
1406     \fi
1407   \else
1408     \eql@tagpos@row@\z@
1409   \fi
1410   \eql@tagpos@prevrow@\z@
1411   \eql@tagpos@headroom@\z@
1412   \eql@tagpos@footroom@\z@
1413 }
```

TODO: describe

```
1414 \def\eql@tagpos@single@eval@continuous{%
1415   \ifnum\eql@tagpos@row@>\z@
1416     \eql@tagpos@target@\eql@tagpos@shift@
1417   \else
1418     \eql@tagpos@target@\dimexpr\eql@line@height@
1419       -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1420   \fi
1421   \eql@tagpos@row@\@ne
1422   \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1423     \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1424     \eql@tagpos@target@\z@
1425   \fi
1426 }
```

TODO: describe

```

1427 \def\eql@tagpos@adjust@eval{%
1428   \if@eqnsw
1429     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1430     \ifnum\eql@tagpos@row@>\eql@totalrows@
1431       \eql@tagpos@row@\eql@totalrows@
1432     \fi
1433     \ifdefined\eql@tagpos@doconvert
1434       \let\eql@tagpos@continuous\eql@true
1435     \fi
1436     \ifdefined\eql@tagpos@continuous
1437       \ifnum\eql@tagpos@row@>\z@
1438         \eql@tagpos@adjust@eval@convert
1439       \fi
1440       \eql@tagpos@adjust@eval@continuous
1441     \fi
1442   \else
1443     \eql@tagpos@row@\z@
1444     \eql@tagpos@prevrow@\z@
1445   \fi
1446 }

```

TODO: describe

```

1447 \def\eql@tagpos@adjust@eval@convert{%
1448   \eql@tagpos@current@\z@
1449   \eql@dimensions@for{%
1450     \ifnum\eql@row@<\eql@tagpos@row@
1451       \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1452         +\eql@line@height@+\eql@line@depth@\relax
1453     \fi
1454     \ifnum\eql@row@=\eql@tagpos@row@
1455       \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1456         +\eql@line@height@-\eql@tagheight@block@\relax
1457     \fi
1458   }%
1459 }

```

TODO: describe

```

1460 \def\eql@tagpos@adjust@eval@continuous{%
1461   \dimen@\dimexpr\eql@tagpos@current@-\eql@tagpos@shift@\relax
1462   \eql@tagpos@row@\eql@totalrows@
1463   \eql@tagpos@prevrow@\z@
1464   \eql@tagpos@headroom@\z@
1465   \eql@tagpos@footroom@\z@
1466   \eql@dimensions@for{%
1467     \ifnum\eql@tagpos@row@=\eql@totalrows@
1468       \eql@tagpos@headroom@\eql@line@interline@
1469       \eql@tagpos@target@\dimexpr\eql@line@interline@
1470         +\eql@line@height@-\dimen@-\eql@tagheight@block@\relax
1471     \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1472       \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1473       \advance\dimen@\eql@tagpos@target@
1474       \eql@tagpos@target@\z@
1475     \fi
1476     \ifdim\eql@tagpos@target@>%
1477       \ifdefined\eql@tagleft-1sp\relax\else\z@\fi
1478       \eql@tagpos@row@\eql@row@
1479       \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1480     \fi

```

```

1481      \advance\dimen@-\dimexpr\eq@line@interline@
1482      +\eq@line@depth@+\eq@line@height@\relax
1483      \fi
1484      \ifnum\eq@row@=\numexpr\eq@tagpos@row@+\@ne\relax
1485      \eq@tagpos@footroom@\eq@line@interline@
1486      \fi
1487    }%
1488  }

```

TODO: describe

```

1489 \def\eq@tagpos@print@line@eval{%
1490   \ifdefined\eq@tagpos@continuous
1491     \eq@tagpos@print@line@eval@continuous
1492   \else
1493     \eq@tagpos@print@line@eval@discrete
1494   \fi
1495 }

```

TODO: describe

```

1496 \def\eq@tagpos@print@line@eval@continuous{%
1497   \if@eqnsw
1498     \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@smashup@\relax
1499     \dp\eq@tagbox@\dimexpr\dp\eq@tagbox@-\eq@tagpos@smashdown@\relax
1500     \eq@tagpos@plain@\eq@tagpos@target@
1501     \@tempdima\dimexpr\eq@line@height@+\eq@tagpos@headroom@
1502     -\ht\eq@tagbox@\relax
1503     \@tempdimb\dimexpr-\eq@line@depth@-\eq@tagpos@footroom@
1504     +\dp\eq@tagbox@\relax
1505     \ifnum\eq@row@=\@ne
1506       \@tempdima.5\maxdimen
1507     \fi
1508     \ifnum\eq@row@=\eq@totalrows@
1509       \@tempdimb-.5\maxdimen
1510     \fi
1511     \ifdim\eq@tagpos@plain@>\@tempdima
1512       \ifdim\eq@tagpos@plain@>\@tempdimb
1513         \ifdim\@tempdima>\@tempdimb
1514           \eq@tagpos@plain@\@tempdima
1515         \else
1516           \eq@tagpos@plain@\@tempdimb
1517         \fi
1518       \fi
1519     \else
1520       \ifdim\eq@tagpos@plain@<\@tempdimb
1521         \ifdim\@tempdima>\@tempdimb
1522           \eq@tagpos@plain@\@tempdimb
1523         \else
1524           \eq@tagpos@plain@\@tempdima
1525         \fi
1526       \fi
1527     \fi
1528     \ifnum\eq@tagpos@prevrow@>\z@
1529       \eq@tagpos@raised@\dimexpr\eq@line@height@+\dp\eq@tagbox@\relax
1530       \ifdim\eq@tagpos@raised@>\eq@tagpos@plain@\else
1531         \eq@tagpos@raised@\eq@tagpos@plain@
1532         \let\eq@tagpos@reserve\eq@false
1533       \fi
1534     \else

```

```

1535 \ifdim\eql@tagpos@target@>%
1536 \ifdefined\eql@tagsleft-1sp\relax\else\z@ \fi
1537 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1538 \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1539 \eql@tagpos@raised@\eql@tagpos@plain@
1540 \let\eql@tagpos@reserve\eql@false
1541 \fi
1542 \else
1543 \eql@tagpos@raised@\dimexpr-\eql@line@depth@
1544 -\ht\eql@tagbox@\relax
1545 \ifdim\eql@tagpos@raised@<\eql@tagpos@plain@\else
1546 \eql@tagpos@raised@\eql@tagpos@plain@
1547 \let\eql@tagpos@reserve\eql@false
1548 \fi
1549 \fi
1550 \fi
1551 \else
1552 \ifnum\eql@tagpos@prevrow@=\eql@row@
1553 \eql@tagwidth@\eql@tagwidth@block@
1554 \else
1555 \let\eql@tagpos@reserve\eql@false
1556 \fi
1557 \fi
1558 }

```

TODO: describe

```

1559 \def\eql@tagpos@print@line@eval@discrete{%
1560 \if@eqnsw
1561 \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1562 \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1563 \eql@tagpos@plain@\eql@tagpos@shift@
1564 \eql@tagpos@headroom@\z@
1565 \eql@tagpos@footroom@\z@
1566 \ifdim\eql@tagpos@shift@>%
1567 \ifdefined\eql@tagsleft-1sp\relax\else\z@ \fi
1568 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1569 \else
1570 \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1571 \fi
1572 \else
1573 \let\eql@tagpos@reserve\eql@false
1574 \fi
1575 }

```

TODO: describe

```

1576 \def\eql@tagpos@print@line@end{%
1577 \ifdefined\eql@tagpos@continuous
1578 \ifnum\eql@tagpos@prevrow@=\eql@row@
1579 \ifdefined\eql@tagpos@reserve
1580 \global\eql@appendexpand\eql@tags@container@block{%
1581 \advance\eql@tagpos@headroom@the\dimexpr\eql@line@height@
1582 +\eql@line@depth@\relax\relax}%
1583 \eql@displaybreak@star\@M
1584 \fi
1585 \fi
1586 \fi
1587 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1588 \let\eql@SK@loaded\eql@false
1589 \let\eql@SK@label\@gobble
1590 \let\eql@SK@clearlabel\@empty
1591 \let\eql@SK@setlabel\@gobble
1592 \let\eql@SK@printlabel@right\@empty
1593 \let\eql@SK@printlabel@left\@empty
1594 \let\eql@SK@printlabel@line\@empty
1595 \def\eql@label@clean{\eql@label@org}
1596 \AddToHook{package/showkeys/after}{%
1597   \let\eql@SK@loaded\eql@true
1598   \def\eql@SK@label#1{\SK@\SK@@label#1}%
1599   \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}%
1600   \eql@SK@clearlabel
1601   \def\eql@SK@@label#1>#2\SK@{%
1602     \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1603   }%
1604   \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}%
1605   \def\eql@SK@printlabel@right{%
1606     \ifx\eql@SK@lab\relax\else
1607       \rlap{\kern\marginparsep\eql@SK@lab}%
1608       \eql@SK@clearlabel
1609     \fi
1610   }%
1611   \def\eql@SK@printlabel@left{%
1612     \ifx\eql@SK@lab\relax\else
1613       \llap{\eql@SK@lab\kern\marginparsep}%
1614       \eql@SK@clearlabel
1615     \fi
1616   }%
1617   \def\eql@SK@printlabel@line{%
1618     \ifx\eql@SK@lab\relax\else
1619       \dimen@\prevdepth
1620       \nointerlineskip
1621       \ifdefined\eql@tagsleft
1622         \llap{%
1623           \eql@SK@lab
1624           \kern\marginparsep
1625         }%
1626         \eql@SK@clearlabel
1627       \else
1628         \rlap{%
1629           \dimen@\displaywidth
1630           \advance\dimen@\marginparsep
1631           \kern\dimen@
1632           \eql@SK@lab
1633         }%
1634       \fi
1635       \eql@SK@clearlabel
1636       \prevdepth\dimen@
1637     \fi
1638   }%
1639   \let\eql@label@org\label
1640   \def\eql@label@clean{\let\SK@\@gobbletwo\eql@label@org}
1641 }

```

Labels.

`eq1@composetag@label` **TODO:** describe

```
1642 \def\eq1@composetag@label{%
1643   \eq1@SK@clearlabel
1644   \ifdefined\eq1@tags@label
1645     \eq1@SK@setlabel\eq1@tags@label
1646     \ifdefined\eq1@tags@name
1647       \let\@currentlabelname\eq1@tags@name
1648     \else
1649       \let\@currentlabelname\eq1@tags@name@generic
1650     \fi
1651     \expandafter\eq1@label@clean\expandafter{\eq1@tags@label}%
1652   \fi
1653 }
```

TODO: describe

```
1654 \def\eq1@tags@printsubeqlabel{%
1655   \eq1@tags@container@parent
1656   \ifdefined\eq1@tags@label
1657     \eq1@tags@makeblockanchor
1658     \eq1@SK@setlabel\eq1@tags@label
1659     \begingroup
1660       \def\@currentcounter{equation}%
1661       \eq1@tags@anchor
1662       \let\@currentlabelname\eq1@tags@name@generic
1663       \protected@edef\@currentlabel{\p@equation\theparentequation}%
1664       \expandafter\eq1@label@clean\expandafter{\eq1@tags@label}%
1665     \endgroup
1666     \eq1@SK@printlabel@line
1667   \fi
1668 }
```

Hyperref Anchors. **TODO:** describe

```
1669 \let\eq1@Hy@anchor\@gobble
1670 \AddToHook{package/hyperref/after}{%
1671   \def\eq1@Hy@anchor#1{%
1672     \Hy@raisedlink{\hyper@anchor{#1}}%
1673   }%
1674 }
```

TODO: describe

```
1675 \def\eq1@tags@makeblockanchor{%
1676   \eq1@tags@glabel@step
1677   \eq1@Hy@anchor\eq1@tags@glabel
1678   \edef\eq1@tags@anchor{%
1679     \def\noexpand\thepage{\thepage}%
1680     \def\noexpand\@currentHref{\eq1@tags@glabel}%
1681   }%
1682 }
```

TODO: describe

`eq1@composetag@anchor`

```
1683 \def\eq1@composetag@anchor{%
```

```

1684 \ifdefined\eql@tags@tag
1685   \def\@currentcounter{equation}%
1686   \ifdefined\eql@tags@ref
1687     \let\@currentlabel\eql@tags@ref
1688   \else
1689     \protected@edef\@currentlabel{\p@equation\eql@tags@tag}%
1690   \fi
1691   \eql@tags@glabel@step
1692   \edef\@currentHref{\eql@tags@glabel}%
1693   \eql@Hy@anchor\@currentHref
1694 \else
1695   \refstepcounter{equation}%
1696   \protected@edef\eql@tags@tag{\theequation}%
1697 \fi
1698 \eql@tags@anchor
1699 }

```

Tag Layout. **TODO:** describe

```

1700 \def\eql@tags@taglayout@set@direct#1{%
1701   \def\eql@tags@taglayout##1{#1}%
1702 }
1703 \def\eql@tags@taglayout@set#1{%
1704   \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1705 }

```

TODO: describe

```

1706 \def\eql@tags@tagform@set@direct#1{%
1707   \def\eql@tags@tagform##1{#1}%
1708 }
1709 \def\eql@tags@tagform@set#1#2#3{%
1710   \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1711 }

1712 \eql@tags@taglayout@set{#1}
1713 \eql@tags@tagform@set({#1})
1714 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1715 \protected\def\tagform{\eql@tags@tagform}
1716 \protected\def\tagbox{\eql@tags@taglayout}
1717 \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:

```

1718 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1719 \def\eql@composetag@tag{%
1720   \eql@tagging@tagbegin
1721   \eql@tags@frame@cmd{%
1722     \eql@tags@taglayout{%
1723       \eql@tags@tagform\eql@tags@tag
1724       \eql@tagging@tagsave
1725     }%
1726   }%
1727   \eql@tagging@tagend
1728 }

```


5.7 Tag Composition

TODO: describe

```

1729 \def\eql@composetag@measure{%
1730   \ifdefined\eql@tags@tag\else
1731     \stepcounter{equation}%
1732     \let\eql@tags@tag\theequation
1733   \fi
1734   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1735   \ifdefined\eql@numbering@multi
1736     \global\let\eql@tags@container\eql@tags@container@clear
1737   \fi
1738 }
```

TODO: describe

```

1739 \def\eql@composetag@print{%
1740   \eql@composetag@anchor
1741   \eql@composetag@label
1742   \ifdefined\eql@tags@left
1743     \eql@SK@printlabel@left
1744     \eql@composetag@tag
1745   \else
1746     \eql@composetag@tag
1747     \eql@SK@printlabel@right
1748   \fi
1749   \global\let\eql@tags@container\eql@tags@container@clear
1750 }
```

TODO: describe

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

```

1751 \def\eql@tagbox@make#1{%
1752   \setbox\eql@tagbox\hbox{\eql@strut@tag\@lign#1}%
1753   \eql@tagwidth@wd\eql@tagbox@
1754   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1755     \eql@tagwidth@\eql@tagwidthmin@
1756   \fi
1757   \advance\eql@tagwidth@\eql@tagsepmin@
1758 }
```

TODO: describe

```

1759 \def\eql@tagbox@print@adjustheadroom{%
1760   \dimen@dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1761   \ifdim\dimen@>\z@
1762     \ifdim\dimen@>\eql@tagpos@headroom@
1763       \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1764     \else
1765       \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1766     \fi
1767   \fi
1768 }
```

TODO: describe

```

1769 \def\eql@tagbox@print@adjustfootroom{%
1770   \dimen@dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1771   \ifdim\dimen@>\z@
1772     \ifdim\dimen@>\eql@tagpos@footroom@
```

```

1773      \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1774      \else
1775      \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1776      \fi
1777      \fi
1778 }

```

TODO: describe

```

1779 \def\eql@tagbox@print@extendabove{%
1780   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1781   \ifdim\dimen@>\z@
1782     \global\eql@appendexpand\eql@display@container{%
1783       \eql@display@aboveextend@the\dimen@\relax}%
1784   \fi
1785 }

```

TODO: describe

```

1786 \def\eql@tagbox@print@extendbelow{%
1787   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1788   \ifdim\dimen@>\z@
1789     \global\eql@appendexpand\eql@display@container{%
1790       \eql@display@belowextend@the\dimexpr\dimen@\relax}%
1791   \fi
1792 }

```

TODO: describe

```

1793 \def\eql@tagbox@print@prepare{%
1794   \ifdefined\eql@tagpos@reserve
1795     \eql@tagpos@current@\eql@tagpos@plain@
1796   \else
1797     \eql@tagpos@current@\eql@tagpos@raised@
1798   \fi
1799   \ifdim\eql@tagpos@headroom@>\z@
1800     \eql@tagbox@print@adjusttheadroom
1801   \fi
1802   \ifdim\eql@tagpos@footroom@>\z@
1803     \eql@tagbox@print@adjustfootroom
1804   \fi
1805   \ifnum\eql@row@=\@ne
1806     \eql@tagbox@print@extendabove
1807   \fi
1808   \ifnum\eql@row@=\eql@totalrows@
1809     \eql@tagbox@print@extendbelow
1810   \fi
1811 }

```

TODO: describe

```

1812 \def\eql@tagbox@print@tagsright{%
1813   \eql@tagbox@print@prepare
1814   \kern-\wd\eql@tagbox@
1815   \raise\eql@tagpos@current@\box\eql@tagbox@
1816 }

```

TODO: describe

```

1817 \def\eql@tagbox@print@tagsleft{%
1818   \eql@display@firstavail@set\z@
1819   \eql@tagbox@print@prepare

```

```

1820 \wd\eql@tagbox@\z@
1821 \raise\eql@tagpos@current@\box\eql@tagbox@
1822 }

```

ql@tagbox@print@cell

```

1823 \def\eql@tagbox@print@cell{%
1824 \eql@tagging@tagaddbox
1825 \ifdefined\eql@tagsleft
1826 \ifdefined\eql@tagpos@reserve
1827 \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1828 \let\eql@tagpos@reserve\eql@false
1829 \fi
1830 \fi
1831 \if@eqnsw
1832 \eql@tagbox@print@tagsleft
1833 \fi
1834 \kern\displaywidth
1835 \else
1836 \kern\displaywidth
1837 \ifdefined\eql@tagpos@reserve
1838 \ifdim\eql@tagwidth@>%
1839 \dimexpr\displaywidth-\eql@line@width@+\eql@tagfuzz@\relax
1840 \let\eql@tagpos@reserve\eql@false
1841 \fi
1842 \fi
1843 \if@eqnsw
1844 \eql@tagbox@print@tagsright
1845 \fi
1846 \fi
1847 }

```

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:

```

1848 \eql@amsmath@undefine\c@parentequation
1849 \eql@amsmath@undefine\theparentequation
1850 \ifdefined\c@parentequation\else
1851 \newcounter{parentequation}
1852 \fi

```

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

```

1853 \def\eql@subequations@template{\theparentequation\alph{equation}}
1854 \def\eql@subequations@template@set#1{\def\eql@tmp##1##2{#1}%
1855 \edef\eql@subequations@template{%
1856 \unexpanded\expandafter{\eql@tmp\theparentequation{equation}}}}

```

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

```
1857 \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`:

```
1858 \def\eq\@subequations@init{%
1859   \edef\eq\@subequations@restorecounter{%
1860     \global\c@equation\the\c@equation\relax}%
1861   \eq\@tags@container@block
1862   \ifdefined\eq\@tags@tag
1863     \eq\@tags@glabel@step
1864     \protected@edef\theHparentequation{\eq\@tags@glabel}%
1865     \protected@edef\theparentequation{\eq\@tags@tag}%
1866   \else
1867     \advance\c@equation\@ne
1868     \protected@edef\theparentequation{\theequation}%
1869     \ifdefined\theHequation
1870       \protected@edef\theHparentequation{\theHequation}%
1871     \fi
1872   \fi
1873   \global\c@parentequation\c@equation
1874   \global\c@equation\z@
1875   \let\theequation\eq\@subequations@template
1876   \def\theHequation{\theHparentequation.\arabic{equation}}}%
1877 }
```

`@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:

```
1878 \def\eq\@subequations@close{%
1879   \ifnum\c@equation=\z@
1880     \eq\@subequations@restorecounter
1881   \else
1882     \global\c@equation\c@parentequation
1883   \fi
1884 }
```

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@printslabel`

```
1885 \def\eq\@subequations@start{%
1886   \let\eq\@tags@container@block\eq\@tags@container@clear
1887   \eq\@nextopt@process{subequations}%
1888   \eq\@subequations@init
1889   \eq\@tags@glabel@step
```

```

1890 \edef\eql@subequations@currentHref{\eql@tags@glabel}%
1891 \eql@Hy@anchor\eql@subequations@currentHref
1892 \edef\eql@subequations@thepage{\thepage}%
1893 \def\@currentcounter{equation}%
1894 \let\@currentHref\eql@subequations@currentHref
1895 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1896 \eql@tags@container@block
1897 \ifdefined\eql@tags@name
1898   \let\@currentlabelname\eql@tags@name
1899 \else
1900   \let\@currentlabelname\eql@tags@name@generic
1901 \fi
1902 \let\eql@subequations@active\eql@true
1903 \ifdefined\eql@tags@label
1904   \eql@SK@label\eql@tags@label
1905 \fi
1906 \ignorespaces
1907 }

```

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

```

1908 \def\eql@subequations@end{%
1909   \ifnum\c@equation>\z@
1910     \eql@tags@container@block
1911     \ifdefined\eql@tags@label
1912       \begingroup
1913         \def\@currentcounter{equation}%
1914         \let\thepage\eql@subequations@thepage
1915         \let\@currentHref\eql@subequations@currentHref
1916         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1917         \ifdefined\eql@tags@name
1918           \let\@currentlabelname\eql@tags@name
1919         \else
1920           \let\@currentlabelname\eql@tags@name@generic
1921         \fi
1922         \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1923       \endgroup
1924     \fi
1925   \fi
1926 \eql@subequations@close
1927 }

```

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

```

1928 \newenvironment{eql@subequations}{%
1929   \eql@verbose@info\eql@verbose@msg@enterenv
1930   \eql@subequations@testall\eql@subequations@start%
1931 }{%
1932   \eql@subequations@end
1933   \ignorespacesafterend
1934   \eql@verbose@info\eql@verbose@msg@leaveenv
1935 }

```

TODO: describe

```

1936 \def\eql@subequations@testall{\eql@parseopt@env\eql@subequations@testall@parse}

```

```

1937 \def\eq@subequations@testall@parse{%
1938   \ifx\eq@parseopt@token[%]
1939     \let\eq@parseopt@next\eq@parseopt@opt
1940   \fi
1941   \ifx\eq@parseopt@token\eq@atxi
1942     \let\eq@parseopt@next\eq@parseopt@label
1943   \fi
1944   \ifx\eq@parseopt@token\eq@atxii
1945     \let\eq@parseopt@next\eq@parseopt@label
1946   \fi
1947   \ifx\eq@parseopt@token\label
1948     \let\eq@parseopt@next\eq@parseopt@end
1949   \fi
1950 }

```

6.3 Subequation Scheme

TODO: describe

```

1951 \def\eq@numbering@subeq@init{%
1952   \let\eq@save@theequation\theequation
1953   \let\eq@save@theHequation\theHequation
1954   \eq@subequations@init
1955   \let\eq@tags@container@parent\eq@tags@container@block
1956   \let\eq@tags@container@block\eq@tags@container@clear
1957 }

```

TODO: describe

```

1958 \def\eq@numbering@subeq@test{%
1959   \ifnum\eq@tagrows@<\tw@
1960     \let\eq@tags@container@block\eq@tags@container@parent
1961     \let\eq@numbering@subeq@use\eq@false
1962     \let\theequation\eq@save@theequation
1963     \let\theHequation\eq@save@theHequation
1964     \eq@subequations@restorecounter
1965   \fi
1966 }

```

TODO: describe

```

1967 % \TODO note must not use setcounter here (when calc is loaded)
1968 \def\eq@numbering@subeq@close{%
1969   \eq@subequations@close
1970 }

```

7 Display Equations Support

TODO: describe

```

1971 \let\eq@display@injectbefore\@undefined
1972 \let\eq@display@injectafter\@undefined
1973 \let\eq@interline@container\@undefined
1974 \def\eq@interline@container@clear{%
1975   \eq@displaybreak@open@\@MM
1976   \eq@vspaceskip@\z@skip
1977 }

```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1978 \interdisplaylinepenalty\@M
```

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

```
1979 \def\eqldgetdsp@pen#1{%
1980   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1981 }
```

TODO: allow a displaybreak before equations

```
1982 \protected\def\eqldisplaybreak@default{%
1983   \eqldwarning{Invalid use of \string\displaybreak}{}%
1984   \eqldteststaropt@loose\@gobble\eqldgobbleopt{}}
1985 \eqldamsmath@after{\let\eqldisplaybreak@default\displaybreak}
1986 \eqldamsmath@let\displaybreak\eqldisplaybreak@default
```

```
1987 \newcount\eqldisplaybreak@pen@
1988 \newcount\eqldisplaybreak@prepen@
1989 \newcount\eqldisplaybreak@postpen@
```

TODO: describe

```
1990 \protected\def\eqldisplaybreak{%
1991   \relax
1992   \eqldampprocttwo\eqldteststaropt@tight
1993   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1994 }
```

```
1995 \def\eqldisplaybreak@star#1{%
1996   \global\eqldappendexpand\eqldinterline@container{%
1997     \eqldisplaybreak@pen@\the\numexpr#1\relax\relax}%
1998 }
```

```
1999 \def\eqldisplaybreak@level[#1]{%
2000   \ifnum#1<\z@
2001     \global\eqldappend\eqldinterline@container{\eqldisplaybreak@pen@\@MM}%
2002   \else
2003     \global\eqldappendexpand\eqldinterline@container{%
2004       \eqldisplaybreak@pen@-\@getpen{#1}\relax}%
2005   \fi
2006 }
```

TODO: describe

```
2007 \def\eqldisplaybreak@pre#1{%
2008   \ifnum#1<\z@
2009     \eqldisplaybreak@prepen@\@MM
2010   \else
2011     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
2012   \fi
2013 }
```

TODO: describe

```
2014 \def\eqldisplaybreak@post#1{%
```

```

2015 \ifnum#1<\z@
2016 \eqldisplaybreak@postpen@MM
2017 \else
2018 \eqldisplaybreak@postpen@-\getpen{#1}\relax
2019 \fi
2020 }

```

TODO: describe

```

2021 \def\eqldisplaybreak@inter#1{%
2022 \ifnum#1<\z@
2023 \interdisplaylinepenaltyM
2024 \else
2025 \interdisplaylinepenalty\eqldspopen{#1}\relax
2026 \fi
2027 }

```

7.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```

2028 \newskip\eqlvspaceskip@

2029 \let\eqlvspace@org\vspace
2030 \def\eqlvspace{%
2031 \ifvmode
2032 \expandafter\eqlvspace@immediate
2033 \else
2034 \expandafter\eqlvspace@line
2035 \fi
2036 }

```

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

```

2037 \def\eqlvspace@line{%
2038 \eq@ifstar@loose\eqlvspace@addfixedbefore\eqlvspace@add
2039 }
2040 \def\eqlvspace@add#1{%
2041 \global\eql@appendexpand\eql@interline@container{%
2042 \advance\eqlvspaceskip@the\glueexpr#1\relax\relax}}
2043 \def\eqlvspace@addfixedbefore#1{%
2044 \global\eql@appendexpand\eql@interline@container{%
2045 \noexpand\eql@append\noexpand\eqldisplay@injectbefore{%
2046 \skip@the\glueexpr#1\relax\relax
2047 \penaltyM
2048 \vskip\skip@
2049 \global\advance\eql@line@interline@\skip@
2050 }%
2051 }%
2052 }
2053 \def\eqlvspace@addfixedafter#1{%
2054 \global\eql@appendexpand\eql@interline@container{%
2055 \noexpand\eql@append\noexpand\eqldisplay@injectafter{%
2056 \dimen@prevdepth
2057 \hrule\@height\z@
2058 \skip@the\glueexpr#1\relax\relax

```



```

2059     \penalty\@M
2060     \vskip\skip@
2061     \global\advance\eql@line@interline@\skip@
2062     \prevdepth\dimen@
2063 }%
2064 }%
2065 }

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

2066 \def\eql@vspace@immediate{%
2067   \noalign\bgroup
2068     \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
2069 }
2070 \def\eql@vspace@fixed#1{%
2071   \skip@\glueexpr#1\relax
2072   \ifnum\eql@row@=\@ne
2073     \global\eql@appendexpand\eql@display@container{%
2074       \advance\eql@abovespace@\the\skip@\relax}%
2075   \else\ifnum\eql@row@>\eql@totalrows@
2076     \global\eql@appendexpand\eql@display@container{%
2077       \advance\eql@belowspace@\the\skip@\relax}%
2078   \else
2079     \dimen@\prevdepth
2080     \hrule\@height\z@
2081     \penalty\@M
2082     \vskip\skip@
2083     \global\advance\eql@line@interline@\skip@
2084     \prevdepth\dimen@
2085   \fi\fi
2086 \egroup
2087 }
2088 \def\eql@vspace@discardable#1{%
2089   \skip@\glueexpr#1\relax
2090   \ifnum\eql@row@=\@ne
2091     \global\eql@appendexpand\eql@display@container{%
2092       \advance\eql@abovespace@\the\skip@\relax}%
2093   \else\ifnum\eql@row@>\eql@totalrows@
2094     \global\eql@appendexpand\eql@display@container{%
2095       \advance\eql@belowspace@\the\skip@\relax}%
2096   \else
2097     \vskip\skip@
2098     \global\advance\eql@line@interline@\skip@
2099   \fi\fi
2100 \egroup
2101 }

```

7.3 Default Vertical Spacing

TODO: describe

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

```

2102 \newbox\eql@strutbox@
2103 \def\eql@strut@depth{.3}
2104 \def\eql@strut{\copy\eql@strutbox@}
2105 \let\eql@strut@cell\eql@strut

```

```

2106 \let\eql@strut@tag\eql@strut
2107 \def\eql@strut@make{%
2108   \setbox\eql@strutbox@\hbox{%
2109     \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
2110     \eql@strut@dimexpr
2111     \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
2112     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
2113   }
2114 }
2115 }
2116 \AtBeginDocument{\eql@strut@make}

```

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

```

2117 \def\eql@spread@set{%
2118   \ifdefined\eql@spread@reset
2119     \lineskip\normalbaselineskip
2120     \lineskiplimit\normallineskiplimit
2121     \baselineskip\normalbaselineskip
2122   \fi
2123   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
2124   +\normalbaselineskip-\baselineskip\relax
2125   \ifdim\eql@spread@>\z@
2126     \openup\eql@spread@
2127     \ifdefined\spread@equation
2128       \let\spread@equation\@empty
2129     \fi
2130   \fi
2131 }

```

7.4 Entry and Exit

TODO: describe

```

2132 \let\eql@beamerbasecolor@apply\@empty
2133 \let\eql@beamerbasecolor@avoid\@empty
2134 \AddToHook{package/beamerbasecolor/after}{%
2135   \def\eql@beamerbasecolor@apply{%
2136     \donotcolorouterdisplaymaths
2137     \donotcoloroutermaths
2138     \beamer@setdisplaymathcolor
2139   }%
2140   \def\eql@beamerbasecolor@avoid{\donotcoloroutermaths}%
2141 }

```

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

```

2142 \newskip\eql@abovespace@
2143 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

2144 \def\eql@display@enter{%
2145   \if@noskipsec\leavevmode\par\fi
2146   \ifvmode
2147     \eql@prevdepth@\prevdepth
2148     \nointerlineskip
2149     \noindent

```

```

2150 \else
2151   \eql@prevdepth@\maxdimen
2152 \fi
2153 \eql@beamerbasecolor@apply
2154 }

```

\eql@display@adjust

```

2155 \def\eql@display@adjust{%
2156   \ifdefined\eql@display@linewidth
2157     \displaywidth\glueexpr\eql@display@linewidth\relax
2158     \advance\displaywidth-\displayindent
2159   \fi
2160   \ifdefined\eql@display@marginleft
2161     \advance\displaywidth\displayindent
2162     \displayindent\glueexpr\eql@display@marginleft\relax
2163     \advance\displaywidth-\displayindent
2164   \fi
2165   \ifdefined\eql@display@marginright
2166     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
2167   \fi
2168   \ifdim\displaywidth<\z@
2169     \displaywidth\z@
2170   \fi
2171 }

```

\eql@display@init

```

2172 \def\eql@display@init{%
2173   \let\eql@display@restore\eql@display@restore@active
2174   \let\displaybreak\eql@displaybreak
2175   \let\eql@vspace@org\vspace
2176   \let\vspace\eql@vspace
2177   \let\eqncontrol\eql@control
2178   \let\eql@display@injectbefore\@empty
2179   \let\eql@display@injectafter\@empty
2180   \let\eqnpunct\eql@punct@setnext
2181   \eql@spread@set
2182   \eql@strut@make
2183   \let\eql@frame@cmd\@undefined
2184 }

```

\eql@display@print

```

2185 \def\eql@display@print{%
2186   \eql@punct@top@set
2187   \let\eql@display@container\@empty
2188   \eql@display@firstavail@\z@
2189   \eql@display@aboveextend@\z@
2190   \eql@display@belowextend@\z@
2191   \global\let\eql@interline@container\eql@interline@container@clear
2192 }

```

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

```

2193 \def\eql@display@halign@init#1{%
2194   \eql@row@\z@
2195   \eql@prevgraf@\prevgraf
2196   \everycr{\noalign{%

```

```

2197 \global\advance\eq\row@\@ne
2198 \prevgraf\numexpr\prevgraf+\@ne\relax
2199 #1%
2200 }}%
2201 }

```

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

```

2202 \def\eq\display@halign@start{%
2203 \prevgraf\numexpr\eq\prevgraf+\@ne\relax
2204 \ifdim\eq\prevdepth@=\maxdimen\else
2205 \prevdepth\eq\prevdepth@
2206 \fi
2207 \ifdim\prevdepth=-\@m\p@\else
2208 \ifdefined\eq\display@height
2209 \skip@\baselineskip
2210 \advance\skip@-\glueexpr\eq\display@height\relax
2211 \advance\skip@-\prevdepth\relax
2212 \ifdim\skip@<\lineskiplimit
2213 \skip@\lineskip
2214 \fi
2215 \advance\skip@-\eq\spread@\relax
2216 \vskip\skip@
2217 \nointerlineskip
2218 \else
2219 \vskip-\eq\spread@\relax
2220 \fi
2221 \fi
2222 }

```

TODO: describe

```

2223 \def\eq\display@vspace{%
2224 \advance\abovedisplayskip\eq\abovespace@
2225 \advance\belowdisplayskip\eq\belowspace@
2226 }

```

TODO: describe

```

2227 \def\eq\display@vspace@native{%
2228 \advance\abovedisplayskip\eq\abovespace@
2229 \advance\belowdisplayskip\eq\belowspace@
2230 \advance\abovedisplayshortskip\eq\abovespace@
2231 \advance\belowdisplayshortskip\eq\belowspace@
2232 }

```

TODO: describe

```

2233 \def\eq\display@penalty{%
2234 \ifnum\eq\displaybreak@postpen@=\@MM\else
2235 \postdisplaypenalty\eq\displaybreak@postpen@
2236 \fi
2237 \ifnum\eq\displaybreak@pen@=\@MM\else
2238 \postdisplaypenalty\eq\displaybreak@pen@
2239 \fi
2240 \ifnum\eq\displaybreak@prepen@=\@MM\else
2241 \predisdisplaypenalty\eq\displaybreak@prepen@
2242 \fi
2243 }

```

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

```

2244 \def\eqldisplay@halign@end{%
2245   \eql@interline@container
2246   \eqldisplay@injectbefore
2247   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
2248   \ifdefined\eqldisplay@depth
2249     \prevdepth\glueexpr\eqldisplay@depth\relax
2250   \fi
2251 }

```

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

```

2252 \def\eqldisplay@close{%
2253   \eqldisplay@container
2254   \ifdim\eqldisplay@firstavail@<\z@
2255     \eqldisplay@firstavail@\z@
2256   \fi
2257   \eql@skip@mode@leave@\z@
2258   \ifdim\eql@prevdepth@=\maxdimen
2259     \ifdim\predisplaysize=-\maxdimen
2260       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
2261       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
2262     \else
2263       \eql@skip@mode@above@\z@
2264       \eql@skip@mode@below@\z@
2265       \advance\eqldisplay@firstavail@\displayindent
2266       \ifdim\eqldisplay@firstavail@>\predisplaysize
2267         \ifcase\eql@skip@mode@short\relax
2268         \or
2269           \eql@skip@mode@above@\@ne
2270         \or
2271           \eql@skip@mode@above@\@ne
2272           \ifnum\eql@totalrows@=\@ne
2273             \eql@skip@mode@below@\@ne
2274           \fi
2275         \or
2276           \eql@skip@mode@above@\@ne
2277           \eql@skip@mode@below@\@ne
2278         \fi
2279       \fi
2280     \fi
2281   \else
2282     \ifdim\eql@prevdepth@=-\@m\p@
2283       \eql@skip@mode@above@\eql@skip@mode@top@above\relax
2284       \eql@skip@mode@below@\eql@skip@mode@top@below\relax
2285     \else
2286       \eql@skip@mode@above@\eql@skip@mode@par@above\relax
2287       \eql@skip@mode@below@\eql@skip@mode@par@below\relax
2288     \fi
2289   \fi
2290   \ifcase\eql@skip@mode@above@
2291   \or\or\or
2292     \predisplaypenalty\z@
2293   \or
2294     \predisplaypenalty\z@
2295   \fi
2296   \ifcase\eql@skip@mode@below@
2297   \or\or\or

```

```

2298 \eq@skip@mode@leave@ \@ne
2299 \or
2300 \eq@skip@mode@leave@ \tw@
2301 \fi
2302 \ifdefined\eq@skip@force@above
2303 \eq@skip@mode@above@ \eq@skip@force@above\relax
2304 \fi
2305 \ifdefined\eq@skip@force@below
2306 \eq@skip@mode@below@ \eq@skip@force@below\relax
2307 \fi
2308 \ifdefined\eq@skip@force@leave
2309 \eq@skip@mode@leave@ \eq@skip@force@leave\relax
2310 \fi
2311 \ifnum\eq@skip@mode@leave@>\z@
2312 \postdisplaypenalty\z@
2313 \fi
2314 \ifcase\eq@skip@mode@above@
2315 \abovedisplayskip\glueexpr\eq@skip@long@above\relax
2316 \or
2317 \abovedisplayskip\glueexpr\eq@skip@short@above\relax
2318 \or
2319 \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
2320 \or
2321 \abovedisplayskip\glueexpr\eq@skip@par@above\relax
2322 \or
2323 \abovedisplayskip\glueexpr\eq@skip@top@above\relax
2324 \or
2325 \abovedisplayskip\z@skip
2326 \or
2327 \abovedisplayskip\glueexpr\eq@skip@med@above\relax
2328 \or
2329 \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
2330 \fi
2331 \ifcase\eq@skip@mode@below@
2332 \belowdisplayskip\glueexpr\eq@skip@long@below\relax
2333 \or
2334 \belowdisplayskip\glueexpr\eq@skip@short@below\relax
2335 \or
2336 \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
2337 \or
2338 \belowdisplayskip\glueexpr\eq@skip@par@below\relax
2339 \or
2340 \belowdisplayskip\glueexpr\eq@skip@top@below\relax
2341 \or
2342 \belowdisplayskip\z@skip
2343 \or
2344 \belowdisplayskip\glueexpr\eq@skip@med@below\relax
2345 \or
2346 \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
2347 \fi
2348 \global\eq@skip@mode@leave@ \eq@skip@mode@leave@
2349 \eq@interline@container
2350 \advance\eq@belowspace@ \eq@vspaceskip@
2351 \eq@display@penalty
2352 \eq@display@vspace
2353 \skip@ \glueexpr\eq@skip@tag@above\relax
2354 \ifdim\skip@>\abovedisplayskip
2355 \skip@ \abovedisplayskip

```

```

2356 \fi
2357 \advance\abovedisplayskip-\eqldisplay@aboveextend@relax
2358 \ifdim\abovedisplayskip<\skip@
2359   \abovedisplayskip\skip@
2360 \fi
2361 \skip@\glueexpr\eql@skip@tag@belowrelax
2362 \ifdim\skip@>\belowdisplayskip
2363   \skip@\belowdisplayskip
2364 \fi
2365 \ifdim\eqldisplay@belowextend@>\z@
2366   \advance\belowdisplayskip-\eqldisplay@belowextend@relax
2367   \ifdim\belowdisplayskip<\skip@
2368     \belowdisplayskip\skip@
2369   \fi
2370 \fi
2371 }

```

TODO: describe

```

2372 \def\eqldisplay@leave{%
2373   \prevgraf\eql@prevgraf@
2374   \ifcase\eql@skip@mode@leave@
2375     \or
2376       \endgraf
2377     \or
2378       \endgraf
2379     \prevdepth-\@m\p@
2380   \fi
2381 }

```

TODO: describe

```

2382 \def\eqldisplay@nest{%
2383   \let\displaybreak\eqldisplaybreak@default
2384   \let\intertext\eql@intertext@default
2385   \let\vspace\eql@vspace@org
2386 }

```

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

```

2387 \def\eqldisplay@restore@active{%
2388   \let\label\eql@label@org
2389   \let>tag\eql>tag@default
2390   \let\raisetag\eql\raisetag@default
2391   \let\displaybreak\eqldisplaybreak@default
2392   \let\intertext\eql@intertext@default
2393   \let\vspace\eql@vspace@org
2394   \ifdefined\eql@amp@mode
2395     \let\&\eql@amp@org
2396   \fi
2397   \let\eqnpunct\eql@punct@adopt
2398   \let\eql@punct@block\@undefined
2399   \let\eqldisplay@restore@\empty
2400 }

```

TODO: describe

```

2401 \let\eqldisplay@restore@\empty
2402 \eql@append\@arrayparboxrestore{%
2403   \eqldisplay@restore
2404   \ifdefined\eql@ampproof@active

```

```

2405 \eql@amprevert
2406 \fi
2407 \@displayfalse
2408 }

```

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)

```

2409 \def\eql@stack@enable{%
2410 \let\eql@stack@save@equations\eql@stack@save@equations@
2411 \let\eql@stack@save@box\eql@stack@save@box@
2412 }

```

TODO: describe

```

2413 \let\eql@stack@save@equations\eql@stack@enable
2414 \let\eql@stack@save@box\eql@stack@enable
2415 \let\eql@stack@restore\@empty

```

TODO: describe

```

2416 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
2417 \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

```

2418 \def\eql@stack@save@equations@{%
2419 \let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2420 \let\eql@stack@cell@container\eql@cell@container
2421 \let\eql@stack@tags@container\eql@tags@container
2422 \let\eql@stack@interline@container\eql@interline@container
2423 \let\eql@stack@dimensions@tab\eql@dimensions@tab
2424 \let\eql@stack@block@container\eql@display@container
2425 \let\eql@stack@punct@top\eql@punct@top
2426 \edef\eql@stack@restore{%
2427 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2428 \eql@stack@save@let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2429 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2430 \eql@stack@save@let\eql@stack@tags@container\eql@tags@container
2431 \eql@stack@save@let\eql@stack@interline@container\eql@interline@container
2432 \eql@stack@save@let\eql@stack@dimensions@tab\eql@dimensions@tab
2433 \eql@stack@save@let\eql@stack@block@container\eql@display@container
2434 \eql@stack@save@let\eql@stack@punct@top\eql@punct@top
2435 \eql@stack@save@reg\eql@column@
2436 \eql@stack@save@reg\eql@totalcolumns@
2437 \eql@stack@save@reg\eql@line@avail@
2438 \eql@stack@save@reg\eql@line@pos@
2439 \eql@stack@save@reg\eql@line@width@
2440 \eql@stack@save@reg\eql@line@depth@
2441 \eql@stack@save@reg\eql@line@height@
2442 \eql@stack@save@reg\eql@line@prevdepth@
2443 \eql@stack@save@reg\eql@line@interline@
2444 \eql@stack@save@reg\eql@totalheight@

```



```

2445 \eqL@stack@save@reg\eqL@tagwidth@max@
2446 \eqL@stack@save@reg\eqL@tagpos@row@
2447 \eqL@stack@save@reg\eqL@row@
2448 \eqL@stack@save@reg\eqL@tagrows@
2449 }%
2450 }

```

TODO: describe

```

2451 \def\eqL@stack@save@box@{%
2452 \let\eqL@stack@cell@container\eqL@cell@container
2453 \edef\eqL@stack@restore{%
2454 \eqL@stack@save@let\eqL@stack@cell@container\eqL@cell@container
2455 \eqL@stack@save@reg\eqL@row@
2456 }%
2457 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2458 \def\eqL@measure@init#1#2{%
2459 \eqL@dimensions@reset
2460 \let\eqL@display@container\@empty
2461 \eqL@numbering@measure@init
2462 \eqL@row@z@
2463 \eqL@totalheight@z@
2464 \eqL@totalrows@M
2465 \eqL@line@prevdepth@-\@m\p@
2466 \eqL@line@interline@z@
2467 \tabskipz@skip
2468 \everycr{\noalign{%
2469 \global\advance\eqL@row@\@ne
2470 #1%
2471 }}%
2472 \eqL@punct@top@set
2473 \global\let\eqL@interline@container\eqL@interline@container@clear
2474 \eqL@measure@savestate
2475 \eqL@multi@cr@let{#2}%
2476 }

```

TODO: describe

```

2477 \def\eqL@measure@tag{%
2478 \eqL@tagwidth@z@
2479 \ifdefined\eqL@numbering@multi
2480 \if@eqnsw
2481 \eqL@tags@container
2482 \eqL@tagbox@make\eqL@composetag@measure
2483 \ifdefined\eqL@tagpos@reserve\else
2484 \eqL@tagwidth@z@
2485 \fi
2486 \fi

```

```

2487 \fi
2488 }

```

TODO: describe

```

2489 \def\eq@measure@endrow{%
2490   \ifdim\eq@line@prevdepth@=-\@m\p@\else
2491     \dimen@\dimexpr\baselineskip-\eq@line@height@-\eq@line@prevdepth@\relax
2492     \ifdim\dimen@<\lineskiplimit
2493       \dimen@\lineskip
2494     \fi
2495     \advance\eq@line@interline@\dimen@
2496   \fi
2497   \eq@dimensions@endrow
2498   \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2499     \global\eq@tagwidth@max@\eq@tagwidth@
2500   \fi
2501   \ifdim\eq@tagwidth@>\z@
2502     \global\advance\eq@tagrows@\@ne
2503   \fi
2504   \global\advance\eq@totalheight@\dimexpr
2505     \eq@line@interline@+\eq@line@height@+\eq@line@depth@
2506   \global\eq@line@interline@\z@
2507   \global\eq@line@prevdepth@\eq@line@depth@
2508 }

```

TODO: describe

```

2509 \def\eq@measure@close{%
2510   \advance\eq@row@-\tw@
2511   \eq@totalrows@\eq@row@
2512   \ifnum\eq@totalrows@>\z@
2513     \eq@dimensions@get\@ne
2514     \eq@topheight@\dimexpr\eq@line@height@+\eq@line@interline@\relax
2515     \eq@dimensions@get\eq@totalrows@
2516     \eq@bottomdepth@\eq@line@depth@
2517   \fi
2518   \eq@numbering@measure@blocktag
2519   \begingroup
2520     \eq@tags@container
2521     \if@eqnsw
2522       \eq@tagbox@make\eq@composetag@measure
2523       \ifdefined\eq@tagpos@reserve\else
2524         \eq@tagwidth@\z@
2525       \fi
2526       \eq@dimensions@saveblocktag
2527     \else
2528       \eq@dimensions@savenoblocktag
2529       \eq@numbering@warnunused
2530     \fi
2531   \endgroup
2532   \eq@dimensions@get\z@
2533   \eq@measure@restorestate
2534 }

```

measure@restorestate

eq@measure@savestate

```

2535 \let\eq@measure@restorestate\@empty
2536 \def\eq@measure@savestate{%
2537   \begingroup

```

```

2538 \def\@elt##1{%
2539 \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2540 \global\edef\@gtempa{\cl@ckpt}%
2541 \endgroup
2542 \let\eq@measure@restorestate\@gtempa
2543 }

```

8.2 Line Breaks

TODO: describe

`\eq@multi@cr`

```

2544 \def\eq@multi@cr{%
2545 \let\eq@punct@term\eq@false
2546 \let\eq@class@rel@composed\@empty
2547 \eq@ampprotect\eq@multi@cr@test\eq@multi@cr@process}

```

TODO: describe

```

2548 \def\eq@multi@cr@test@setopt{%
2549 \let\eq@multi@cr@test\eq@multi@cr@testopt}
2550 \def\eq@multi@cr@test@setall{%
2551 \let\eq@multi@cr@test\eq@multi@cr@testall}

```

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

```

2552 \def\eq@multi@cr@testopt#1{\eq@teststaropt@tight
2553 {\eq@displaybreak@star\@M\eq@multi@cr@testopt@set{#1}}}%
2554 {\eq@multi@cr@testopt@set{#1}}{Opt}}
2555 \def\eq@multi@cr@testopt@set#1[#2]{\eq@vspace@add{#2}#1}

```

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

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

```

2556 \def\eq@multi@cr@testall{\eq@parseopt@cr\eq@multi@cr@testall@parse}
2557 \def\eq@multi@cr@testall@parse{%
2558 \ifx\eq@parseopt@token[%
2559 \let\eq@parseopt@next\eq@multi@cr@parse@vspace
2560 \fi
2561 \ifx\eq@parseopt@token*%
2562 \let\eq@parseopt@next\eq@multi@cr@parse@star
2563 \fi
2564 \ifx\eq@parseopt@token.%
2565 \let\eq@parseopt@next\eq@parseopt@punctpass
2566 \fi
2567 \ifx\eq@parseopt@token,%
2568 \let\eq@parseopt@next\eq@parseopt@punctpass
2569 \fi
2570 \ifx\eq@parseopt@token~%
2571 \let\eq@parseopt@next\eq@parseopt@punctpass
2572 \fi
2573 \ifx\eq@parseopt@token'%
2574 \let\eq@parseopt@next\eq@parseopt@punctnext
2575 \fi
2576 \ifx\eq@parseopt@token!%
2577 \let\eq@parseopt@next\eq@parseopt@punctterm
2578 \fi
2579 \ifx\eq@parseopt@token/%

```

```

2580 \let\eql@parseopt@next\eql@multi@cr@parse@break
2581 \fi
2582 \ifx\eql@parseopt@token=%
2583 \let\eql@parseopt@next\eql@parseopt@relsymp
2584 \fi
2585 \ifx\eql@parseopt@token;%
2586 \let\eql@parseopt@next\eql@parseopt@relcont
2587 \fi
2588 \ifx\eql@parseopt@token:~
2589 \let\eql@parseopt@next\eql@parseopt@relstart
2590 \fi
2591 \ifx\eql@parseopt@token|~
2592 \let\eql@parseopt@next\eql@parseopt@relord
2593 \fi
2594 \ifx\eql@parseopt@token?~
2595 \let\eql@parseopt@next\eql@multi@cr@parse@rel
2596 \fi
2597 \ifx\eql@parseopt@token&~
2598 \let\eql@parseopt@next\eql@parseopt@end
2599 \fi
2600 }
2601 \def\eql@multi@cr@parse@vspace[#1]{\eql@vspace@add{#1}\eql@parseopt@peek}
2602 \def\eql@multi@cr@parse@star#1{\eql@displaybreak@star\@M\eql@parseopt@peek}
2603 \def\eql@multi@cr@parse@break{\numbernext\eql@parseopt@punctclear}
2604 \def\eql@multi@cr@parse@rel#1#2{%
2605 \def\eql@tmp{#2}%
2606 \ifx\eql@tmp\eql@relax\else
2607 \ifdefined\eql@punct@next\else
2608 \eql@punct@next@clear
2609 \fi
2610 \ifdefined\eql@multi@cr@relnext\numbernext\fi
2611 \fi
2612 \ifdefined\eql@multi@linesmode
2613 \ifx\eql@tmp\@empty
2614 \def\eql@class@rel@composed{\eql@shape@cont}%
2615 \else
2616 \def\eql@class@rel@composed{\eql@shape@rel#2}%
2617 \fi
2618 \else
2619 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
2620 \fi
2621 \eql@parseopt@end}

```

eql@multi@cr@process

```

2622 \def\eql@multi@cr@process{%
2623 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
2624 \edef\eql@tmp{%
2625 \unexpanded{%
2626 \eql@multi@endline
2627 \cr
2628 \eql@multi@cr@interline
2629 }%
2630 \unexpanded\expandafter{\eql@class@rel@composed}%
2631 }%
2632 \eql@tmp
2633 }

2634 \def\eql@multi@cr@interline{%

```

```

2635 \noalign{%
2636   \eq@interline@container
2637   \eq@display@injectbefore
2638   \ifnum\eq@displaybreak@pen@=\@MM
2639     \penalty\interdisplaylinepenalty
2640   \else
2641     \penalty\eq@displaybreak@pen@
2642   \fi
2643   \vskip\eq@vspaceskip@
2644   \global\advance\eq@line@interline@\eq@vspaceskip@
2645   \eq@display@injectafter
2646   \global\let\eq@interline@container\eq@interline@container@clear
2647 }%
2648 }

```

`\eq@multi@cr@let`

```

2649 \def\eq@multi@cr@let#1{%
2650   \let\\\eq@multi@cr
2651   \let\eq@multi@endline#1%
2652 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2653 \def\eq@intertext@default{\eq@error{Invalid use of \string\intertext}}
2654 \eq@amsmath@let\intertext\eq@intertext@default

```

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

```

2655 \def\eq@intertext@process{%
2656   \eq@multi@endline
2657   \cr
2658   \ifmeasuring@
2659     \expandafter\@gobble
2660   \else
2661     \expandafter\eq@intertext@print
2662   \fi
2663 }

```

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

```

2664 \def\eq@intertext@print#1{%
2665   \noalign{%
2666     \eq@display@halign@end
2667     \let\eq@skip@force@below\z@
2668     \let\eq@skip@force@above\z@
2669     \eq@setkeys{intertext}\eq@intertext@opt
2670     \openup-\eq@spread@
2671     \penalty\postdisplaypenalty
2672     \ifcase\eq@skip@force@below\relax
2673       \advance\eq@vspaceskip@\glueexpr\eq@skip@long@below\relax
2674     \or
2675       \advance\eq@vspaceskip@\glueexpr\eq@skip@short@below\relax
2676     \or
2677       \advance\eq@vspaceskip@\glueexpr\eq@skip@cont@below\relax

```

```

2678 \or
2679 \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2680 \or
2681 \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2682 \or
2683 \advance\eql@vspaceskip@\z@skip
2684 \or
2685 \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2686 \or
2687 \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2688 \fi
2689 \vskip\eql@vspaceskip@
2690 \global\let\eql@interline@container\eql@interline@container@clear
2691 \vbox{%
2692 \@parboxrestore
2693 \ifdim
2694 \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2695 \else
2696 \parshape\@ne
2697 \@totalleftmargin\linewidth
2698 \fi
2699 \noindent
2700 \prevgraf\eql@prevgraf@
2701 \ignorespaces
2702 #1%
2703 \par
2704 \global\eql@prevgraf@\prevgraf
2705 }%
2706 \penalty\predisplaypenalty
2707 \ifcase\eql@skip@force@above\relax
2708 \vskip\glueexpr\eql@skip@long@above\relax
2709 \or
2710 \vskip\glueexpr\eql@skip@short@above\relax
2711 \or
2712 \vskip\glueexpr\eql@skip@cont@above\relax
2713 \or
2714 \vskip\glueexpr\eql@skip@par@above\relax
2715 \or
2716 \vskip\glueexpr\eql@skip@top@above\relax
2717 \or
2718 \vskip\z@skip
2719 \or
2720 \vskip\glueexpr\eql@skip@med@above\relax
2721 \or
2722 \vskip\glueexpr\eql@skip@custom@above\relax
2723 \fi
2724 % \eql@prevdepth@\maxdimen
2725 \eql@prevdepth@\z@
2726 \eql@display@halign@start
2727 }
2728 }

```

TODO: describe

```

2729 \newenvironment{eql@intertext}{%
2730 \eql@testopt@tight\eql@intertext@}%
2731 }{%
2732 \aftergroup\eql@intertext@after
2733 \ignorespacesafterend

```

2734 }

TODO: describe

```
2735 \def\eql@intertext@env{intertext}
2736 \def\eql@intertext@[#1]{%
2737   \global\def\eql@intertext@opt{#1}%
2738   \ifx\@currenvir\eql@intertext@env
2739     \def\eql@scan@call{\eql@intertext@inject\eql@scan@end}%
2740     \expandafter\eql@scan@env
2741   \else
2742     \expandafter\eql@intertext@process
2743   \fi
2744 }
```

TODO: describe

```
2745 \def\eql@intertext@inject{%
2746   \global\edef\eql@intertext@after{%
2747     \noexpand\eql@intertext@process{%
2748       \ifx\eql@scan@body\eql@scan@body@dump
2749         \eql@scan@body@dump
2750       \else
2751         \noexpand\scantokens{\eql@scan@body@dump}%
2752       \fi
2753     }%
2754   }%
2755 }
```

8.4 Line Marks

TODO: describe

```
2756 \def\eql@markline@pos@below{below}
2757 \def\eql@markline@pos@bottom{bottom}
2758 \def\eql@markline@pos@baseline{baseline}
2759 \let\eql@markline@pos\eql@markline@pos@baseline
2760 \let\eql@markline@shift\z@
2761 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2762 \def\eql@markline@symbol{}
```

TODO: describe

```
2763 \def\eql@markline@select#1{%
2764   \let\eql@markline@shift\z@
2765   \eql@setkeys{markline}{#1}%
2766   \eql@markline@print
2767 }
```

TODO: describe

```
2768 \def\eql@markline@print{%
2769   \dimen@=\dimexpr\eql@markline@shift\relax
2770   \ifx\eql@markline@pos\eql@markline@pos@below
2771     \ifdim\dimen@=\z@\else
2772       \penalty\@M
2773       \vskip-\dimen@
2774     \fi
2775     \nointerlineskip
2776     \penalty\@M
```

```

2777 \vbox{\hfill\hbox{\eq@markline@symbol}}}%
2778 \else
2779 \ifx\eq@markline@pos\eq@markline@pos@baseline
2780 \advance\dimen@ \prevdepth
2781 \fi
2782 \setbox\z@\hbox{\raise\dimen@\hbox{\eq@markline@symbol}}}%
2783 \dimen@ \prevdepth
2784 \ht\z@\z@
2785 \dp\z@\z@
2786 \nointerlineskip
2787 \penalty\@M
2788 \vbox{\hfill\box\z@}%
2789 \prevdepth\dimen@
2790 \fi
2791 }

```

TODO: describe

```

2792 \def\eq@markline@inject#1{%
2793 \let\eq@markline@push\eq@false
2794 \ifx\eq@markline@pos\eq@markline@pos@below\else
2795 \ifdefined\eq@tagsleft\else
2796 \ifx\eq@equations@main\eq@multi@main
2797 \ifdefined\eq@numbering@multi
2798 \if@eqnsw
2799 \let\eq@markline@push\eq@true
2800 \fi
2801 \else
2802 \ifnum\eq@row@=\eq@tagpos@row@
2803 \let\eq@markline@push\eq@true
2804 \fi
2805 \fi
2806 \else
2807 \if@eqnsw
2808 \let\eq@markline@push\eq@true
2809 \fi
2810 \fi
2811 \fi
2812 \fi
2813 \ifdefined\eq@markline@push
2814 \global\eq@append\eq@interline@container{%
2815 \eq@append\eq@display@injectbefore{\eq@markline@select{push,#1}}}%
2816 \else
2817 \global\eq@append\eq@interline@container{%
2818 \eq@append\eq@display@injectbefore{\eq@markline@select{#1}}}%
2819 \fi
2820 }

```

TODO: describe

```

2821 \def\eq@markline@amsthm@opt[#1]{\eq@markline@inject{qed,#1}}
2822 \def\eq@markline@amsthm@staropt[#1]{\eq@markline@inject{qed,push,#1}}
2823 \def\eq@markline@amsthm@qed{\eq@teststaropt@tight
2824 \eq@markline@amsthm@staropt\eq@markline@amsthm@opt{}}
2825 \def\eq@markline@amsthm@register#1{\eq@letcs{#1@qed}\eq@markline@amsthm@qed}
2826 \def\eq@markline@amsthm@move#1#2{%
2827 \AddToHook{package/amsthm/after}{%
2828 \eq@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:

```
2829 \newcount\eql@shape@pos@
2830 \newdimen\eql@shape@amount@
2831 \let\eql@shape@lastrow\eql@false
```

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

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

```
2832 \newdimen\eql@marginleft@
2833 \newdimen\eql@marginright@
2834 \newdimen\eql@marginleft@min@
2835 \newdimen\eql@centeroffset@
```

9.2 Shape Schemes

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

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

```
2836 \def\eql@shape@tab@default{default}
2837 \def\eql@shape@tab@left{left}
2838 \def\eql@shape@tab@center{center}
2839 \def\eql@shape@tab@right{right}
2840 \def\eql@shape@tab@first{first}
2841 \def\eql@shape@tab@hanging{hanging}
2842 \def\eql@shape@tab@steps{steps}
```

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

```
2843 \let\eql@shape@tab@def\eql@shape@tab@default
2844 \let\eql@shape@tab@\eql@shape@tab@default
2845 \let\eql@shape@tab@l\eql@shape@tab@left
2846 \let\eql@shape@tab@c\eql@shape@tab@center
2847 \let\eql@shape@tab@r\eql@shape@tab@right
2848 \let\eql@shape@tab@rc\eql@shape@tab@right
2849 \let\eql@shape@tab@indent\eql@shape@tab@first
2850 \let\eql@shape@tab@f\eql@shape@tab@first
2851 \let\eql@shape@tab@hang\eql@shape@tab@hanging
2852 \let\eql@shape@tab@lc\eql@shape@tab@hanging
2853 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
2854 \let\eql@shape@tab@h\eql@shape@tab@hanging
2855 \let\eql@shape@tab@lcr\eql@shape@tab@steps
2856 \let\eql@shape@tab@s\eql@shape@tab@steps
```

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

```
2857 \let\eql@shape@mode\eql@shape@tab@default
```

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

```

2858 \def\eq@shape@set#1{%
2859   \ifcsname eq@shape@tab@#1\endcsname
2860     \expandafter\let\expandafter\eq@shape@mode
2861       \csname eq@shape@tab@#1\endcsname
2862   \else
2863     \eq@error{shape '#1' unknown: setting to default}%
2864     \let\eq@shape@mode\eq@shape@tab@default
2865   \fi
2866 }

```

`\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:

```

2867 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@ \eq@shape@amount@z@}
2868 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ \ne \eq@shape@amount@z@}
2869 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@ \eq@shape@amount@z@}
2870 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2871 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@ \eq@shape@amount@z@}
2872 \def\eq@shape@layoutleft@center{\eq@shape@pos@ \ne \eq@shape@amount@z@}
2873 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@ \eq@shape@amount@z@}
2874 \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):

```

2875 \def\eq@shape@layoutcenter@first{%
2876   \eq@shape@pos@z@
2877   \eq@shape@amount@z@
2878   \ifnum\eq@totalrows@>\@ne
2879     \ifnum\eq@row@=\@ne
2880       \eq@shape@amount@\eq@indent@
2881     \fi
2882   \fi
2883 }
2884 \def\eq@shape@layoutleft@first{%
2885   \eq@shape@pos@z@
2886   \eq@shape@amount@z@
2887   \ifnum\eq@totalrows@>\@ne
2888     \ifnum\eq@row@=\@ne
2889       \eq@shape@amount@\eq@indent@
2890     \fi
2891   \fi
2892 }

```

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:

```

2893 \def\eq@shape@layoutcenter@hanging{%
2894   \eq@shape@pos@z@
2895   \eq@shape@amount@\eq@indent@
2896   \ifnum\eq@totalrows@>\@ne
2897     \ifnum\eq@row@=\@ne
2898       \eq@shape@amount@z@
2899     \fi
2900   \fi
2901 }

```

```

2902 \def\eq\@shape@layoutleft@hanging{%
2903   \eq\@shape@pos@\z@
2904   \eq\@shape@amount@\z@
2905   \ifnum\eq\@totalrows@>\@ne
2906     \ifnum\eq\@row@=\@ne
2907       \eq\@shape@amount@-\eq\@indent@
2908     \fi
2909   \fi
2910 }

```

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:

```

2911 \def\eq\@shape@layoutcenter@steps{%
2912   \eq\@shape@amount@\z@
2913   \eq\@shape@pos@\@ne
2914   \ifnum\eq\@totalrows@>\@ne
2915     \ifnum\eq\@row@=\@ne
2916       \eq\@shape@pos@\z@
2917     \fi
2918     \ifnum\eq\@row@=\eq\@totalrows@
2919       \eq\@shape@pos@\tw@
2920     \fi
2921   \fi
2922 }
2923 \def\eq\@shape@layoutleft@steps{%
2924   \eq\@shape@pos@\z@
2925   \eq\@shape@amount@\z@
2926   \ifnum\eq\@totalrows@>\@ne
2927     \ifnum\eq\@row@=\@ne
2928       \eq\@shape@amount@-\eq\@indent@
2929     \fi
2930     \ifnum\eq\@row@=\eq\@totalrows@
2931       \eq\@shape@amount@\eq\@indent@
2932     \fi
2933   \fi
2934 }

```

`\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`:

```

2935 \let\eq\@shape@eval\@undefined
2936 \def\eq\@shape@select{%
2937   \expandafter\let\expandafter\eq\@shape@eval
2938   \csname eq\@shape%
2939     @\ifdefined\eq\@layoutleft layoutleft\else layoutcenter\fi
2940     @\eq\@shape@mode\endcsname
2941 }

```

`\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`

```

2942 \protected\def\eq\@shape@alignleft{%
2943   \global\eq\@append\eq\@cell@container{\eq\@shape@pos@\z@}%
2944   \eq\@ampprotect\eq\@shape@align@testpar\eq\@shape@alignamount@opt}
2945 \protected\def\eq\@shape@aligncenter{%
2946   \global\eq\@append\eq\@cell@container{\eq\@shape@pos@\@ne}%
2947   \eq\@ampprotect\eq\@shape@align@testpar\eq\@shape@alignamount@opt}

```

```

2948 \protected\def\eq@shape@alignright{%
2949   \global\eq@append\eq@cell@container{\eq@shape@pos@tw}%
2950   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2951 \def\eq@shape@align@testpar#1{%
2952   \eq@ifstar@tight{#1[\eq@indent@]}%
2953   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2954   {\eq@testopt@tight{#1}\z@}}
2955 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2956 \protected\def\eq@shape@alignamount{%
2957   \eq@ampprotecttwo\eq@ifstar@tight
2958   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2959 \def\eq@shape@alignamount@add#1{%
2960   \global\eq@appendexpand\eq@cell@container{%
2961     \advance\eq@shape@amount@the\glueexpr#1\relax\relax}}
2962 \def\eq@shape@alignamount@set#1{%
2963   \global\eq@appendexpand\eq@cell@container{%
2964     \eq@shape@amount@the\glueexpr#1\relax\relax}}
2965 \def\eq@shape@align@enable{%
2966   \let\shoveleft\eq@shape@alignleft
2967   \let\shovecenter\eq@shape@aligncenter
2968   \let\shoveright\eq@shape@alignright
2969   \let\shoveby\eq@shape@alignamount
2970 }

```

TODO: describe

```

2971 \protected\def\eq@shape@align@default{%
2972   \eq@warn@here{\shove...}%
2973   \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2974 \protected\def\eq@shape@alignamount@default{%
2975   \eq@warn@here{\shove...}%
2976   \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2977 \def\eq@shape@align@disable{%
2978   \let\shoveleft\eq@shape@align@default
2979   \let\shovecenter\eq@shape@align@default
2980   \let\shoveright\eq@shape@align@default
2981   \let\shoveby\eq@shape@alignamount@default
2982 }

```

9.3 Width Data

width@block@ (*dimen*)

```

2983 \newdimen\eq@tagwidth@block@
2984 \newdimen\eq@tagheight@block@
2985 \newdimen\eq@tagdepth@block@

```

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

```

2986 \let\eq@dimensions@tab\@empty

```

eq@dimensions@reset

```

2987 \def\eq@dimensions@reset{%
2988   \let\eq@dimensions@tab\@empty
2989   \eq@tagwidth@max@\z@

```

```

2990 \eql@tagrows@\z@
2991 }

\eql@dimensions@add

2992 \def\eql@dimensions@add#1{%
2993 \global\eql@appendexpand\eql@dimensions@tab{#1}%
2994 }

eql@dimensions@addreg

2995 \def\eql@dimensions@addreg#1{#1\the#1\relax}

@dimensions@startrow

2996 \def\eql@dimensions@startrow{%
2997 \eql@dimensions@add{\eql@dimensions@addreg\eql@row@}%
2998 }

@dimensions@savecell

2999 \def\eql@dimensions@savecell{%
3000 \eql@dimensions@add{%
3001 \eql@dimensions@addreg\eql@shape@pos@
3002 \eql@dimensions@addreg\eql@cellwidth@
3003 \eql@dimensions@addreg\eql@shape@amount@
3004 \noexpand\eql@dimensions@cellcall
3005 }%
3006 }

l@dimensions@savesep

3007 \def\eql@dimensions@savesep{%
3008 \eql@dimensions@add{\noexpand\eql@dimensions@sepcall}%
3009 }

eql@dimensions@endrow

3010 \def\eql@dimensions@endrow{%
3011 \eql@dimensions@add{,%
3012 \eql@dimensions@addreg\eql@tagwidth@
3013 \eql@dimensions@addreg\eql@line@height@
3014 \eql@dimensions@addreg\eql@line@depth@
3015 \eql@dimensions@addreg\eql@line@interline@
3016 };%
3017 }

ensions@saveblocktag

3018 \def\eql@dimensions@saveblocktag{%
3019 \eql@dimensions@add{\eql@row@0\relax,%
3020 \eql@tagwidth@block@\the\eql@tagwidth@\relax
3021 \eql@tagheight@block@\the\ht\eql@tagbox@\relax
3022 \eql@tagdepth@block@\the\dp\eql@tagbox@\relax
3023 \eql@dimensions@addreg\eql@tagpos@shift@
3024 \let\noexpand\eql@tagpos@reserve\ifdefined\eql@tagpos@reserve
3025 \noexpand\eql@true\else\noexpand\eql@false\fi
3026 };%
3027 \global\eql@tagwidth@max@\eql@tagwidth@
3028 \global\eql@tagrows@\@ne
3029 }

```

sions@savenoblocktag

```
3030 \def\eqldimensions@savenoblocktag{%
3031   \eqldimensions@add{\eqldimensions@row@0\relax,;}%
3032 }
```

\eqldimensions@for

```
3033 \def\eqldimensions@for#1{%
3034   \def\eqldimensions@forall{#1}%
3035   \expandafter\eqldimensions@forstep\eqldimensions@tab
3036 }
```

l@dimensions@forstep

```
3037 \def\eqldimensions@forstep\eqldimensions@row@#1\relax#2,#3{%
3038   \eqldimensions@row@#1\relax
3039   \ifnum\eqldimensions@row@#1=\z@ \else
3040     #3%
3041     \def\eqldimensions@cells{#2}%
3042     \eqldimensions@forall
3043     \expandafter\eqldimensions@forstep
3044   \fi
3045 }
```

\eqldimensions@get

```
3046 \def\eqldimensions@get#1{%
3047   \eqldimensions@row@#1\relax
3048   \expandafter\eqldimensions@getdef\expandafter{\the\eqldimensions@row@}%
3049   \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil
3050 }
```

ql@dimensions@getdef

```
3051 \def\eqldimensions@getdef#1{%
3052   \def\eqldimensions@getparse
3053     ##1\eqldimensions@row@#1\relax##2,##3;##4\@nil{%
3054     ##3%
3055     \def\eqldimensions@cells{##2}%
3056   }%
3057 }
```

\eqldimensions@colwidth@tab

```
3058 \let\eqldimensions@colwidth@tab\empty
```

\eqldimensions@colwidth@get

```
3059 \def\eqldimensions@colwidth@get#1{%
3060   \ifcase\expandafter#1\eqldimensions@colwidth@tab\else\z@ \fi
3061 }
```

\eqldimensions@colwidth@save

```
3062 \def\eqldimensions@colwidth@save#1{%
3063   \edef\eqldimensions@colwidth@tab{%
3064     \noexpand\or\the#1%
3065     \unexpanded\expandafter{\eqldimensions@colwidth@tab}%
3066   }%
3067 }
```

`\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@`:

```

3068 \def\eql@dimensions@calc{%
3069   \eql@column@\z@
3070   \eql@line@pos@\z@
3071   \eql@line@possep@\z@
3072   \eql@line@avail@\eql@totalwidth@
3073   \eql@line@availsep@\eql@intercolumns@
3074   \eql@line@width@\z@
3075   \eql@line@widthsep@\z@
3076   \let\eql@dimensions@cellcall\eql@dimensions@calc@call
3077   \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
3078   \eql@dimensions@cells
3079 }

```

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

```

3080 \def\eql@dimensions@calc@callsep{%
3081   \advance\eql@line@possep@\@ne
3082 }%

```

`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?!)

```

3083 \def\eql@dimensions@calc@call{%
3084   \advance\eql@column@\@ne
3085   \ifnum\eql@totalcolumns@=\@ne
3086     \dimen@\eql@totalwidth@
3087   \else
3088     \dimen@\eql@colwidth@get\eql@column@\relax
3089   \fi
3090   \ifdim\eql@cellwidth@>\z@
3091     \ifdim\eql@line@width@=\z@
3092       \eql@line@avail@\eql@line@pos@
3093       \eql@line@availsep@\eql@line@possep@
3094       \ifcase\eql@shape@pos@
3095       \or
3096         \advance\eql@line@avail@\dimexpr
3097           (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
3098       \or
3099         \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
3100       \fi
3101       \advance\eql@line@avail@\eql@shape@amount@
3102     \fi
3103     \eql@line@width@\eql@line@pos@
3104     \eql@line@widthsep@\eql@line@possep@
3105     \ifcase\eql@shape@pos@
3106     \advance\eql@line@width@\eql@cellwidth@
3107   \or
3108     \advance\eql@line@width@\dimexpr
3109       (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
3110   \or
3111     \advance\eql@line@width@\dimen@

```

```

3112     \fi
3113     \advance\eql@line@width@\eql@shape@amount@
3114 \fi
3115 \advance\eql@line@pos@\dimen@
3116 }

```

9.4 Best Line Selection

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

```

3117 \let\eql@numbering@best@auto\eql@false

```

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

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

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

```

3118 \newcount\eql@numbering@best@row@
3119 \newdimen\eql@numbering@best@space@
3120 \let\eql@numbering@best@use\eql@false

```

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

```

3121 \def\eql@numbering@best@find{%
3122   \eql@numbering@best@row@ \z@
3123   \eql@numbering@best@space@ \z@
3124   \eql@dimensions@for{%
3125     \eql@dimensions@calc
3126     \ifdefined\eql@tagsleft
3127       \dimen@\eql@line@avail@
3128     \else
3129       \dimen@\dimexpr\eql@totalwidth@-\eql@line@width@\relax
3130     \fi
3131     \ifdim\dimen@>\eql@numbering@best@space@
3132       \eql@numbering@best@row@\eql@row@
3133       \eql@numbering@best@space@\dimen@
3134     \fi
3135   }%
3136   \ifnum\eql@numbering@best@row@>\z@
3137     \eql@tagpos@row@\eql@numbering@best@row@
3138     \let\eql@tagpos@continuous\eql@false
3139     \eql@tagpos@prevrow@\z@
3140   \fi
3141 }

```

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

```

3142 \def\eql@numbering@best@test#1{%
3143   \eql@dimensions@get#1%
3144   \eql@dimensions@calc
3145   \ifdefined\eql@tagsleft
3146     \dimen@\dimexpr\eql@line@avail@
3147       +\eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax
3148   \else
3149     \dimen@\dimexpr\displaywidth-\eql@line@width@
3150       -\eql@marginleft@-\eql@line@widthsep@\eql@colsep@\relax
3151   \fi
3152   \ifdim\dimen@<\eql@tagwidth@block@
3153     \let\eql@numbering@best@use\eql@true
3154   \fi
3155 }

```


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

```

3156 \def\eq@numbering@best@eval{%
3157   \ifdefined\eq@numbering@best@auto
3158     \ifdefined\eq@numbering@best@use\else
3159       \ifdefined\eq@numbering@multi\else
3160         \ifnum\eq@tagpos@row@>\z@
3161           \eq@numbering@best@test\eq@tagpos@row@
3162         \fi
3163         \ifnum\eq@tagpos@prevrow@>\z@
3164           \eq@numbering@best@test\eq@tagpos@prevrow@
3165         \fi
3166       \fi
3167     \fi
3168   \fi
3169   \ifdefined\eq@numbering@best@use
3170     \eq@numbering@best@find
3171   \fi
3172 }

```

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`

```

3173 \def\eq@adjust@calc@tagmargin{%
3174   \ifdefined\eq@tagmargin@val
3175     \eq@tagmargin@\glueexpr\eq@tagmargin@val\relax
3176   \else
3177     \eq@tagmargin@\eq@tagwidth@max@
3178     \ifdim\eq@tagmargin@>\z@
3179       \advance\eq@tagmargin@-\eq@tagsepmin@
3180     \fi
3181   \fi

3182   \dimen@\eq@tagrows@\p@
3183   \ifnum\eq@totalrows@=\@one
3184     \ifnum\eq@tagrows@=\@one
3185       \advance\dimen@1sp\relax
3186     \fi
3187   \fi
3188   \ifdim\dimen@>\eq@totalrows@\eq@tagmargin@ratio@\else
3189     \eq@tagmargin@\z@
3190   \fi

3191   \@tempdima\dimexpr\displaywidth
3192     -\eq@totalwidth@-\eq@intercolumns@\eq@colsepmin@\relax
3193   \@tempdimb\dimexpr\@tempdima-\tw@\eq@tagmargin@\relax
3194   \ifdim\@tempdimb>\z@
3195     \ifdim\eq@tagmargin@threshold\@tempdima<\@tempdimb
3196       \eq@tagmargin@\z@
3197     \fi
3198   \fi
3199 }

```

9.6 Single Column

ql@adjust@calc@lines

```

3200 \def\eql@adjust@calc@lines{%
3201   \eql@totalcolumns@\ne
3202   \eql@intercolumns@\z@
3203   \eql@colsep@\z@
3204   \ifdefined\eql@layoutleft
3205     \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
3206     \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
3207     \ifdim\eql@marginleft@<\eql@marginleft@min@
3208       \eql@marginleft@\eql@marginleft@min@
3209     \fi
3210     \dimen@\glueexpr\eql@layoutleftmarginmax\relax
3211     \ifdim\eql@marginleft@>\dimen@
3212       \eql@marginleft@\dimen@
3213     \fi
3214     \eql@marginright@\z@
3215     \eql@centeroffset@\z@
3216   \else
3217     \eql@adjust@calc@tagmargin
3218     \ifdefined\eql@paddingleft@val
3219       \eql@marginleft@\dimexpr
3220         (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
3221         -\glueexpr\eql@paddingleft@val\relax\relax
3222       \ifdim\eql@marginleft@<\z@
3223         \eql@marginleft@\z@
3224       \fi
3225     \else
3226       \eql@marginleft@\z@
3227     \fi
3228     \ifdefined\eql@paddingright@val
3229       \eql@marginright@\dimexpr
3230         (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
3231         -\glueexpr\eql@paddingright@val\relax\relax
3232       \ifdim\eql@marginright@<\z@
3233         \eql@marginright@\z@
3234       \fi
3235     \else
3236       \eql@marginright@\z@
3237     \fi
3238     \ifdim\eql@tagmargin@>\z@
3239       \ifdefined\eql@tagsleft
3240         \ifdim\eql@marginleft@<\eql@tagsepmin@
3241           \eql@marginleft@\eql@tagsepmin@
3242         \fi
3243         \advance\eql@marginleft@\eql@tagmargin@
3244         \advance\eql@centeroffset@\eql@tagmargin@
3245       \else
3246         \ifdim\eql@marginright@<\eql@tagsepmin@
3247           \eql@marginright@\eql@tagsepmin@
3248         \fi
3249         \advance\eql@marginright@\eql@tagmargin@
3250         \advance\eql@centeroffset@-\eql@tagmargin@
3251       \fi
3252     \fi
3253     \eql@marginleft@min@\z@
3254     \eql@centeroffset@\dimexpr\eql@marginright@-\eql@marginleft@

```

```

3255         \ifdefined\eql@tagsleft+\else-\fi\eql@tagmargin@relax
3256     \fi

3257     \eql@totalwidth@\dimexpr\displaywidth
3258         -\eql@marginleft@-\eql@marginright@relax
3259 }

```

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:

```

3260 \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:

```

3261     \ifnum\eql@totalcolumns@<\thr@@
3262         \eql@totalcolumns@\tw@
3263         \let\eql@columns@fulllength\eql@false
3264     \fi

```

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

```

3265     \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@relax

```

Evaluate the minimum intercolumn space which we will need often:

```

3266     \eql@colsepmin@\glueexpr\eql@colsepmin@valrelax

```

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

```

3267     \ifdefined\eql@layoutleft
3268         \eql@marginleft@\glueexpr\eql@layoutleftmargin@relax
3269         \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin@relax
3270         \ifdim\eql@marginleft@<\eql@marginleft@min@
3271             \eql@marginleft@\eql@marginleft@min@
3272         \fi
3273     \else

```

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

```

3274         \eql@adjust@calc@tagmargin
3275         \ifdefined\eql@columns@fulllength
3276             \ifdim\eql@tagmargin@>\z@
3277                 \advance\eql@tagmargin@\eql@tagsepmin@
3278             \fi
3279         \fi
3280         \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
3281             -\eql@intercolumns@\eql@colsepmin@relax
3282             \eql@tagmargin@\z@
3283         \fi
3284         \eql@marginleft@min@\z@
3285     \fi

```

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

```
3286 \ifnum\eq@intercolumns@>\z@
```

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

```
3287 \eq@colsep@\dimexpr\displaywidth-\eq@totalwidth@\relax
3288 \ifdefined\eq@layoutleft
3289 \advance\eq@colsep@-\eq@marginleft@
3290 \else
3291 \advance\eq@colsep@-\eq@tagmargin@
3292 \fi
3293 \count@\eq@intercolumns@
3294 \ifdefined\eq@columns@fulllength\else
3295 \ifdefined\eq@layoutleft
3296 \advance\count@\@ne
3297 \else
3298 \advance\count@\tw@
3299 \fi
3300 \fi
3301 \divide\eq@colsep@\count@
```

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

```
3302 \ifdim\eq@colsep@<\eq@colsepmin@
3303 \eq@colsep@\eq@colsepmin@
3304 \else
3305 \ifdefined\eq@columns@fulllength\else
3306 \dimen@\glueexpr\eq@colsepmax@val\relax
3307 \ifdim\eq@colsep@>\dimen@
3308 \eq@colsep@\dimen@
3309 \fi
3310 \fi
3311 \fi
3312 \else
```

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

```
3313 \eq@colsep@\eq@colsepmin@
3314 \fi
```

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

```
3315 \ifdefined\eq@layoutleft
```

Set the default value:

```
3316 \ifdim\eq@colsep@=\eq@colsepmin@
```

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

```
3317 \dimen@\dimexpr\displaywidth-\eq@totalwidth@
3318 -\eq@intercolumns@\eq@colsep@\relax
3319 \ifdim\dimen@<\eq@marginleft@
3320 \ifdim\dimen@<\eq@marginleft@min@
3321 \eq@marginleft@\eq@marginleft@min@
3322 \else
```

```

3323         \eql@marginleft@\dimen@
3324     \fi
3325 \fi
3326 \fi
3327 \else

```

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

```

3328     \ifdefined\eql@columns@fulllength
3329         \eql@marginleft@z@

```

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

```

3330 \else
3331     \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@
3332         -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
3333     \ifdim\eql@marginleft@<z@
3334         \eql@marginleft@z@
3335     \fi
3336 \fi

```

Add tag margin in case of left tags:

```

3337     \ifdefined\eql@tagsleft
3338         \advance\eql@marginleft@\eql@tagmargin@
3339     \fi
3340 \fi

```

Find the best row for tag placement:

```

3341 \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:

```

3342 \ifdefined\eql@tagsleft
3343     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
3344 \else
3345     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
3346 \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

```

3347 \ifdefined\eql@numbering@multi
3348     \eql@dimensions@for{%
3349         \ifdim\eql@tagwidth@>z@
3350             \eql@dimensions@calc
3351             \eql@adjust@columns@test
3352         \fi
3353     }%
3354 \else
3355     \ifnum\eql@tagpos@row@>z@
3356         \ifnum\eql@tagpos@row@>\eql@totalrows@\else
3357             \eql@dimensions@get\eql@tagpos@row@
3358             \eql@tagwidth@\eql@tagwidth@block@
3359             \eql@dimensions@calc
3360             \eql@adjust@columns@test
3361         \fi

```

```

3362 \fi
3363 \ifnum\eq@tagpos@prevrow@>\z@
3364 \eq@dimensions@get\eq@tagpos@prevrow@
3365 \eq@tagwidth@\eq@tagwidth@block@
3366 \eq@dimensions@calc
3367 \eq@adjust@columns@test
3368 \fi
3369 \fi

```

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

```

3370 \advance\eq@totalwidth@\dimexpr
3371 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
3372 }

```

Placement for Right Tags.

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

```

3373 \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?

```

3374 \@tempdima\dimexpr\displaywidth-\eq@line@width@-\eq@tagwidth@\relax

```

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

```

3375 \ifdim\@tempdima<\dimexpr
3376 \eq@marginleft@+\eq@line@widthsep@\eq@colsep@\relax

```

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

```

3377 \ifdim\@tempdima<\dimexpr
3378 \eq@marginleft@min@+\eq@line@widthsep@\eq@colsepmin@\relax\else

```

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

```

3379 \eq@adjust@columns@modify@tagsright
3380 \fi
3381 \fi
3382 }

```

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

```

3383 \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.

```

3384 \ifnum\eq@line@widthsep@>\z@
3385 \dimen@\@tempdima
3386 \count@\eq@line@widthsep@
3387 \ifdefined\eq@layoutleft
3388 \advance\dimen@-\eq@marginleft@

```

```

3389 \else
3390 \ifdefined\eql@columns@fulllength\else
3391 \advance\count@\@ne
3392 \fi
3393 \fi
3394 \divide\dimen@\count@

```

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

```

3395 \ifdim\dimen@<\eql@colsep@
3396 \ifdim\dimen@<\eql@colsepmin@
3397 \eql@colsep@\eql@colsepmin@
3398 \else
3399 \eql@colsep@\dimen@
3400 \fi
3401 \fi
3402 \fi

```

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

```

3403 \dimen@\dimexpr\@tempdima-\eql@line@widthsep@\eql@colsep@\relax
3404 \ifdim\eql@marginleft@>\dimen@
3405 \eql@marginleft@\dimen@
3406 \fi
3407 }

```

Placement for Left Tags.

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

```

3408 \def\eql@adjust@columns@test@tagsleft{%

```

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

```

3409 \count@\numexpr\eql@intercolumns@-\eql@line@availsep@\relax
3410 \@tempdima\dimexpr\eql@tagwidth@-\eql@line@avail@\relax

```

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

```

3411 \ifdim\@tempdima>\dimexpr
3412 \eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax

```

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

```

3413 \ifdim\eql@tagwidth@<%
3414 \ifdefined\eql@layoutleft
3415 \glueexpr\eql@layoutleftmarginmax\relax
3416 \else
3417 \displaywidth
3418 \fi

```

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

```

3419 \ifdim\@tempdima>\dimexpr
3420 \displaywidth-\eql@totalwidth@-\count@\eql@colsepmin@\relax\else

```

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

```

3421      \eq@adjust@columns@modify@tagsleft
3422      \fi
3423      \fi
3424      \fi
3425 }

```

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

```

3426 \def\eq@adjust@columns@modify@tagsleft{%

```

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

```

3427 \ifnum\count@>\z@
3428   \dimen@ \dimexpr\displaywidth-\eq@totalwidth@-\@tempdima\relax
3429   \ifdefined\eq@columns@fulllength\else
3430     \advance\count@\@ne
3431   \fi
3432   \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.

```

3433   \ifdim\dimen@<\eq@colsep@
3434     \ifdim\dimen@<\eq@colsepmin@
3435       \dimen@\eq@colsepmin@
3436     \fi
3437     \advance\dimen@-\eq@colsep@
3438     \advance\eq@marginleft@-\eq@intercolumns@\dimen@
3439     \advance\eq@colsep@\dimen@
3440   \fi
3441 \fi

```

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

```

3442   \dimen@ \dimexpr\@tempdima-\eq@line@availsep@\eq@colsep@\relax
3443   \ifdim\eq@marginleft@<\dimen@
3444     \eq@marginleft@\dimen@
3445   \fi
3446 }

```

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:

```

3447 \ifdefined\inf@bad\else%
3448   \newcount\inf@bad
3449   \inf@bad1000000\relax
3450 \fi

```


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

```
3451 \let\eq restore@hfuzz\empty
3452 \def\eq save@hfuzz{\edef\eq restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

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

```
3453 \newcount\eq alignbadness@
3454 \newcount\eq tagbadness@
3455 \newcount\eq arrange@badness@
3456 \eq alignbadness@\inf@bad
3457 \eq tagbadness@\inf@bad
```

10.2 Arrangement Methods

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

```
3458 \def\eq arrange@try#1{%
3459   \ifdim#1>\dimexpr\displaywidth-\eq cellwidth@\relax
3460     \setbox\eq cellbox@\hbox to\displaywidth{%
3461       \unhbox\eq cellbox@\unkern\kern#1}%
3462     \eq arrange@badness@\badness
3463   \else
3464     \eq arrange@badness@\m@ne
3465   \fi
3466 }
```

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

TODO: adjust

```
3467 \def\eq arrange@print#1#2{%
3468   \eq restore@hfuzz
3469   \if@eqnsw
3470     \ifdefined\eq tagsleft
3471       \eq tagbox@print@tagsleft
3472     \fi
3473   \fi
3474   \hbox to\displaywidth{%
3475     #1%
3476     \unhbox\eq cellbox@\unkern
3477     #2%
3478     \eq tagging@mathaddlast
3479   }%
3480   \if@eqnsw
3481     \ifdefined\eq tagsleft\else
3482       \eq tagbox@print@tagsright
3483     \fi
3484   \fi
3485 }
```

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

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

```

3486 \def\eql@arrange@print@alignleft#1#2{%
3487   \eql@display@firstavail@set{\dimexpr#1\relax}%
3488   \eql@arrange@print{\kern#1}{\kern#2}%
3489 }

3490 \def\eql@arrange@print@alignright#1#2{%
3491   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3492   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3493 }

3494 \def\eql@arrange@print@aligncenter#1{%
3495   \eql@display@firstavail@set{\dimexpr
3496     (\displaywidth-\eql@cellwidth@+1)/\tw@\relax}%
3497   \ifdim#1>\z@
3498     \eql@arrange@print{\kern#1\hfil}{}%
3499   \else
3500     \eql@arrange@print{\hfil}{\kern-#1}%
3501   \fi
3502 }
```

`\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:

```

3503 \def\eql@arrange@init{%
3504   \eql@save@hfuzz
3505   \hfuzz\maxdimen
3506   \eql@shape@select
3507 }
```

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

```

3508 \def\eql@arrange@print@line{%
3509   \eql@tagging@tagaddbox
3510   \csname eql@arrange%
3511     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3512     @init\endcsname
3513   \csname eql@arrange%
3514     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3515     @\ifdefined\eql@tagpos@reserve
3516       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3517       notag\fi\endcsname
3518 }
```

10.3 Central Alignment

TODO: describe

```

3519 \def\eql@arrange@aligncenter@init{%
3520   \eql@tagging@aligncenter
3521   \eql@line@offset@dimexpr\tw@\eql@shape@amount@
3522     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3523 }
```

TODO: describe

```

3524 \def\eql@arrange@aligncenter@notag{%
3525   \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3526     \ifdim\eql@line@offset@<\eql@marginleft@min@
3527       \dimexpr\tw@eql@marginleft@min@-\eql@line@offset@>\relax
3528     \else
3529       \eql@line@offset@
3530     \fi
3531   \eql@arrange@print@aligncenter\eql@line@offset@
3532 \else
3533   \ifdim\eql@line@offset@<\eql@marginleft@min@
3534     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3535   \else
3536     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3537   \fi
3538 \fi
3539 }

```

TODO: describe

```

3540 \def\eql@arrange@aligncenter@tagsright{%
3541   \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3542     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@>\relax
3543       \dimexpr\tw@eql@marginleft@min@-\eql@line@offset@>\relax
3544     \else
3545       \dimexpr\tw@eql@tagwidth@+\eql@line@offset@>\relax
3546     \fi
3547   \eql@arrange@print@aligncenter\eql@line@offset@
3548 \else
3549   \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@>\relax}%
3550   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3551     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@>\relax
3552       \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3553     \else
3554       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3555     \fi
3556   \else
3557     \let\eql@tagpos@reserve\eql@false
3558     \eql@arrange@aligncenter@notag
3559   \fi
3560 \fi
3561 }

```

```

3562 \def\eql@arrange@aligncenter@tagsleft{%
3563   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3564     \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3565       \ifdim\eql@line@offset@<\eql@tagwidth@
3566         \dimexpr\tw@eql@tagwidth@-\eql@line@offset@>\relax
3567       \else
3568         \eql@line@offset@
3569       \fi
3570     \eql@arrange@print@aligncenter\eql@line@offset@
3571 \else
3572   \eql@arrange@try\eql@tagwidth@
3573   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3574     \ifdim\eql@line@offset@<\eql@tagwidth@
3575       \eql@arrange@print@alignleft\eql@tagwidth@\z@
3576     \else
3577       \eql@arrange@print@alignright\eql@tagwidth@\z@

```

```

3578     \fi
3579   \else
3580     \let\eql@tagpos@reserve\eql@false
3581     \eql@arrange@aligncenter@notag
3582   \fi
3583 \fi
3584 \else
3585   \eql@arrange@aligncenter@notag
3586 \fi
3587 }

```

10.4 Left Alignment

```

3588 \def\eql@arrange@alignleft@init{%
3589   \eql@tagging@alignleft
3590   \eql@line@offset@{\dimexpr\eql@marginleft@+\eql@shape@amount@\relax
3591   \ifdim\eql@line@offset@<\eql@marginleft@min@
3592     \eql@line@offset@{\eql@marginleft@min@
3593   \fi
3594 }

3595 \def\eql@arrange@alignleft@notag{%
3596   \ifdim\eql@line@offset@>\eql@marginleft@min@
3597     \eql@arrange@try\eql@line@offset@
3598     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3599       \eql@arrange@print@alignleft\eql@line@offset@\z@
3600     \else
3601       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3602     \fi
3603   \else
3604     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3605   \fi
3606 }

3607 \def\eql@arrange@alignleft@tagsright{%
3608   \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3609   \ifnum\eql@arrange@badness@<\eql@alignbadness@
3610     \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3611   \else
3612     \ifdim\eql@line@offset@>\eql@marginleft@min@
3613       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3614     \fi
3615     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3616       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3617     \else
3618       \let\eql@tagpos@reserve\eql@false
3619       \eql@arrange@alignleft@notag
3620     \fi
3621   \fi
3622 }

3623 \def\eql@arrange@alignleft@tagsleft{%
3624   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3625     \ifdim\eql@line@offset@>\eql@tagwidth@
3626       \eql@arrange@try\eql@line@offset@
3627     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3628       \eql@arrange@print@alignleft\eql@line@offset@\z@
3629     \else
3630       \eql@arrange@try\eql@tagwidth@
3631     \ifnum\eql@arrange@badness@<\eql@tagbadness@

```

```

3632         \eql@arrange@print@alignright\eql@tagwidth@z@
3633     \else
3634         \let\eql@tagpos@reserve\eql@false
3635         \eql@arrange@print@alignright\eql@marginleft@min@z@
3636     \fi
3637 \fi
3638 \else
3639     \eql@arrange@try\eql@tagwidth@
3640     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3641         \eql@arrange@print@alignleft\eql@tagwidth@z@
3642     \else
3643         \let\eql@tagpos@reserve\eql@false
3644         \eql@arrange@alignleft@notag
3645     \fi
3646 \fi
3647 \else
3648     \eql@arrange@alignleft@notag
3649 \fi
3650 }

```

10.5 Right Alignment

```

3651 \def\eql@arrange@alignright@init{%
3652     \eql@tagging@alignright
3653     \eql@line@offset@\dimexpr\eql@marginright@-\eql@shape@amount@\relax
3654     \ifdim\eql@line@offset@<z@
3655         \eql@line@offset@z@
3656     \fi
3657 }

```

TODO: describe

```

3658 \def\eql@arrange@alignright@notag{%
3659     \ifdim\eql@line@offset@>z@
3660         \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3661         \ifnum\eql@arrange@badness@<\eql@alignbadness@
3662             \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3663         \else
3664             \eql@arrange@print@alignleft\eql@marginleft@min@z@
3665         \fi
3666     \else
3667         \eql@arrange@print@alignright\eql@marginleft@min@z@
3668     \fi
3669 }

```

TODO: describe

```

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

```

```

3684 \else
3685   \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3686   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3687     \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3688   \else
3689     \let\eql@tagpos@reserve\eql@false
3690     \eql@arrange@alignright@notag
3691   \fi
3692 \fi
3693 }

```

TODO: describe

```

3694 \def\eql@arrange@alignright@tagsleft{%
3695   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3696     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3697     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3698       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3699     \else
3700       \ifdim\eql@line@offset@>\z@
3701         \eql@arrange@try\eql@tagwidth@
3702       \fi
3703       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3704         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3705       \else
3706         \let\eql@tagpos@reserve\eql@false
3707         \eql@arrange@alignright@notag
3708       \fi
3709     \fi
3710   \else
3711     \eql@arrange@alignright@notag
3712   \fi
3713 }

```

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`

```

3714 \def\eql@box@cr{%
3715   \ifmmode\else\unskip\fi
3716   \eql@vspaceskip@\z@skip
3717   \let\eql@punct@term\eql@false
3718   \let\eql@class@rel@composed\@empty
3719   \eql@ampprotect\eql@box@cr@test\eql@box@cr@process
3720 }

```

TODO: describe

```

3721 \def\eql@box@cr@test@setopt{\let\eql@box@cr@test\eql@box@cr@testopt}
3722 \def\eql@box@cr@test@setall{\let\eql@box@cr@test\eql@box@cr@testall}

```

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

```

3723 \def\eql@box@cr@testopt#1{\eql@teststaropt@tight
3724   {\eql@box@cr@testopt@set{#1}}{\eql@box@cr@testopt@set{#1}}{0pt}}
3725 \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

```

3726 \def\eql@box@cr@testall{\eql@parseopt@cr\eql@box@cr@testall@parse}
3727 \def\eql@box@cr@testall@parse{%
3728   \ifx\eql@parseopt@token[%
3729     \let\eql@parseopt@next\eql@parseopt@vspace
3730   \fi
3731   \ifx\eql@parseopt@token*%
3732     \let\eql@parseopt@next\eql@parseopt@gobble
3733   \fi
3734   \ifx\eql@parseopt@token.%
3735     \let\eql@parseopt@next\eql@parseopt@punctpass
3736   \fi
3737   \ifx\eql@parseopt@token,%
3738     \let\eql@parseopt@next\eql@parseopt@punctpass
3739   \fi
3740   \ifx\eql@parseopt@token~%
3741     \let\eql@parseopt@next\eql@parseopt@punctpass
3742   \fi
3743   \ifx\eql@parseopt@token'%
3744     \let\eql@parseopt@next\eql@parseopt@punctnext
3745   \fi
3746   \ifx\eql@parseopt@token!%
3747     \let\eql@parseopt@next\eql@parseopt@punctterm
3748   \fi
3749   \ifx\eql@parseopt@token/%
3750     \let\eql@parseopt@next\eql@parseopt@punctclear
3751   \fi
3752   \ifx\eql@parseopt@token=%
3753     \let\eql@parseopt@next\eql@parseopt@relsymp
3754   \fi
3755   \ifx\eql@parseopt@token;%
3756     \let\eql@parseopt@next\eql@parseopt@relcont
3757   \fi
3758   \ifx\eql@parseopt@token:%
3759     \let\eql@parseopt@next\eql@parseopt@relstart
3760   \fi
3761   \ifx\eql@parseopt@token|%
3762     \let\eql@parseopt@next\eql@parseopt@relord
3763   \fi
3764   \ifx\eql@parseopt@token?%
3765     \let\eql@parseopt@next\eql@box@cr@parse@rel
3766   \fi
3767   \ifx\eql@parseopt@token&%
3768     \let\eql@parseopt@next\eql@parseopt@end
3769   \fi
3770 }
3771 \def\eql@box@cr@parse@rel#1#2{%
3772   \def\eql@tmp{#2}%
3773   \ifx\eql@tmp\eql@relax\else

```

```

3774 \ifdefined\eql@punct@next\else
3775 \eql@punct@next@clear
3776 \fi
3777 \fi
3778 \ifx\eql@box@open\eql@box@lines@open
3779 \ifx\eql@tmp@empty
3780 \def\eql@class@rel@composed{\eql@shape@cont}%
3781 \else
3782 \def\eql@class@rel@composed{\eql@shape@rel#2}%
3783 \fi
3784 \fi
3785 \ifx\eql@box@open\eql@box@columns@open
3786 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3787 \fi
3788 \ifx\eql@box@open\eql@box@cases@open
3789 \def\eql@class@rel@composed{&#2}%
3790 \fi
3791 \eql@parseopt@end}

```

`\eql@box@cr@process`

```

3792 \def\eql@box@cr@process{%
3793 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
3794 \edef\eql@tmp{%
3795 \unexpanded{%
3796 \eql@box@endline
3797 \eql@box@lastcell
3798 \cr
3799 }%
3800 \noalign{%
3801 \vskip\the\eql@vspaceskip@relax
3802 }%
3803 \unexpanded\expandafter{\eql@class@rel@composed}%
3804 }%
3805 \eql@tmp
3806 }

```

`\eql@box@endline`

```

3807 \def\eql@box@endline{%
3808 \eql@punct@apply@line
3809 \eql@hook@lineout
3810 }

```

11.2 Column Breaks

TODO: describe

```

3811 \def\eql@box@amp{%
3812 \eql@ampprotecttwo\eql@box@amp@testescape\eql@amp@org
3813 \eql@box@amp@process}
3814 \def\eql@box@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
3815 \relax
3816 \let\eql@punct@term\eql@false
3817 \let\eql@class@rel@composed\empty
3818 \eql@box@amp@test{#2}}}

```

TODO: describe


```

3819 \def\eql@box@amp@test@setopt{%
3820   \let\eql@box@amp@test\eql@box@amp@testopt}
3821 \def\eql@box@amp@test@setall{%
3822   \let\eql@box@amp@test\eql@box@amp@testall}

\eql@box@amp@testopt TODO: describe

3823 \let\eql@box@amp@testopt\@empty

\eql@box@amp@testall TODO: describe
ox@amp@testall@parse
3824 \def\eql@box@amp@testall{\eql@parseopt@cr\eql@box@amp@testall@parse}
3825 \def\eql@box@amp@testall@parse{%
3826   \ifx\eql@parseopt@token.%
3827     \let\eql@parseopt@next\eql@parseopt@punctpass
3828   \fi
3829   \ifx\eql@parseopt@token,%
3830     \let\eql@parseopt@next\eql@parseopt@punctpass
3831   \fi
3832   \ifx\eql@parseopt@token~%
3833     \let\eql@parseopt@next\eql@parseopt@punctpass
3834   \fi
3835   \ifx\eql@parseopt@token'%
3836     \let\eql@parseopt@next\eql@parseopt@punctnext
3837   \fi
3838   \ifx\eql@parseopt@token!%
3839     \let\eql@parseopt@next\eql@parseopt@punctterm
3840   \fi
3841   \ifx\eql@parseopt@token=%
3842     \let\eql@parseopt@next\eql@parseopt@relsymp
3843   \fi
3844   \ifx\eql@parseopt@token;%
3845     \let\eql@parseopt@next\eql@parseopt@relcont
3846   \fi
3847   \ifx\eql@parseopt@token:%
3848     \let\eql@parseopt@next\eql@parseopt@relstart
3849   \fi
3850   \ifx\eql@parseopt@token|%
3851     \let\eql@parseopt@next\eql@parseopt@relord
3852   \fi
3853   \ifx\eql@parseopt@token?%
3854     \let\eql@parseopt@next\eql@box@amp@parse@rel
3855   \fi
3856   \ifx\eql@parseopt@token&%
3857     \let\eql@parseopt@next\eql@parseopt@end
3858   \fi
3859 }
3860 \def\eql@box@amp@parse@rel#1#2{%
3861   \ifx\eql@box@open\eql@box@columns@open
3862     \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3863   \else
3864     \def\eql@class@rel@composed{#2}%
3865   \fi
3866   \eql@parseopt@end}

\eql@box@amp@process

3867 \def\eql@box@amp@process{%
3868   \ifdefined\eql@punct@term\eql@punct@apply@top\fi

```

```

3869 \ifx\eql@box@open\eql@box@columns@open
3870 \edef\eql@tmp{%
3871 \ifx\eql@class@rel@composed\@empty
3872 \ifx\eql@box@lastcell\eql@box@columns@lastcell@odd
3873 &\noexpand\eql@punct@next@clear\fi&%
3874 \else
3875 \ifx\eql@box@lastcell\eql@box@columns@lastcell@even&\fi%
3876 \unexpanded\expandafter{\eql@class@rel@composed}%
3877 \fi
3878 }%
3879 \else
3880 \edef\eql@tmp{%
3881 \ifx\eql@class@rel@composed\@empty
3882 &%
3883 \else
3884 \unexpanded\expandafter{\eql@class@rel@composed}%
3885 \fi
3886 }%
3887 \fi
3888 \eql@tmp
3889 }

```

11.3 Lines Mode

```

3890 \def\eql@box@lines@lastcell{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3891 \def\eql@box@lines@open{%
3892 \eql@shape@align@enable
3893 \let\eql@box@lastcell\eql@box@lines@lastcell
3894 \everycr{\noalign{%
3895 \eql@verbose@info\eql@verbose@msg@startline@number
3896 \global\advance\eql@row@\@ne
3897 }}%
3898 \tabskip\z@skip
3899 \halign\bgroup
3900 &%
3901 \global\let\eql@cell@container\@empty
3902 \setbox\eql@cellbox@\hbox{%
3903 \eql@strut@cell
3904 \@lign
3905 $\m@th\eql@mathstyle
3906 \eql@hook@colin
3907 ##%
3908 \eql@punct@apply@col
3909 \eql@hook@colout
3910 \eql@tagging@mathsave
3911 $%
3912 \eql@tagging@mathaddlast
3913 }%
3914 \ifdefined\eql@shape@lastrow
3915 \eql@totalrows@\eql@row@
3916 \fi
3917 \eql@shape@eval
3918 \eql@cell@container
3919 \ifdefined\eql@frame@cmd
3920 \ifcase\eql@shape@pos@
3921 \eql@frame@measure

```

```

3922         \advance\eql@shape@amount@-\eql@frame@margin@
3923     \or\or
3924         \eql@frame@measure
3925         \advance\eql@shape@amount@+\eql@frame@margin@
3926     \fi
3927     \eql@frame@print
3928 \fi
3929 \ifcase\eql@shape@pos@
3930     \kern\eql@shape@amount@
3931     \box\eql@cellbox@
3932     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3933         -\eql@shape@amount@+\@flushglue\relax
3934     \eql@tagging@alignleft
3935 \or
3936     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3937     \box\eql@cellbox@
3938     \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3939     \eql@tagging@aligncenter
3940 \or
3941     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3942         +\eql@shape@amount@+\@flushglue\relax
3943     \box\eql@cellbox@
3944     \kern-\eql@shape@amount@
3945     \eql@tagging@alignright
3946 \fi
3947     \tabskip\eql@colsep@\relax
3948 \crrc
3949 \noalign{%
3950     \global\let\eql@shape@lastrow\eql@false
3951     \eql@hook@blockbefore
3952 }%
3953 \eql@hook@blockin
3954 }
3955 \def\eql@box@lines@set{\let\eql@box@open\eql@box@lines@open}

```

11.4 Columns Mode

```

3956 \def\eql@box@columns@lastcell@odd{%
3957     &\omit
3958     \eql@prevwidth@\wd\eql@cellbox@
3959     \let\eql@frame@cmd\eql@frame@prevcmd
3960     \ifdefined\eql@frame@cmd
3961         \eql@frame@measure
3962         \advance\eql@prevwidth@\eql@frame@margin@
3963     \eql@frame@print
3964 \fi
3965     \kern-\eql@prevwidth@
3966     \unhbox\eql@cellbox@
3967     \hfil
3968     &\omit\kern-\eql@colsep@
3969 }%
3970 \def\eql@box@columns@lastcell@even{&\omit\kern-\eql@colsep@}
3971 \def\eql@box@columns@open{%
3972 % \TODO templates
3973     \eql@shape@align@disable
3974     \let\eql@box@lastcell@empty
3975     \everycr{\noalign{%
3976         \eql@verbose@info\eql@verbose@msg@startline@new

```

```

3977 }}%
3978 \tabskip\z@skip
3979 \halign\bgroup
3980   &%
3981   \let\eql@box@lastcell\eql@box@columns@lastcell@odd
3982   \global\let\eql@cell@container\@empty
3983   \global\setbox\eql@cellbox@\hbox{%
3984     \eql@strut@cell
3985     \@lign
3986     $\m@th\eql@mathstyle
3987     \eql@hook@colin
3988     ##%
3989     \eql@punct@apply@next
3990     \eql@class@innerleft
3991     \eql@hook@innerleft
3992     \eql@tagging@mathsave
3993     $%
3994     \eql@tagging@mathaddlast
3995   }%
3996   \eql@cell@container
3997   \hfil
3998   \kern\wd\eql@cellbox@
3999   \ifdefined\eql@frame@cmd
4000     \eql@frame@measure
4001     \kern\eql@frame@margin@
4002   \fi
4003   \global\let\eql@frame@prevcmd\eql@frame@cmd
4004   \tabskip\z@skip
4005   &%
4006   \eql@prevwidth@\wd\eql@cellbox@
4007   \let\eql@box@lastcell\eql@box@columns@lastcell@even
4008   \let\eql@frame@cmd\eql@frame@prevcmd
4009   \global\let\eql@cell@container\@empty
4010   \setbox\eql@cellbox@\hbox{%
4011     \unhbox\eql@cellbox@
4012     \eql@strut@cell
4013     \@lign
4014     $\m@th\eql@mathstyle
4015     \eql@hook@innerright
4016     \eql@class@innerright@sel
4017     ##%
4018     \eql@punct@apply@col
4019     \eql@hook@colout
4020     \eql@tagging@mathsave
4021     $%
4022     \eql@tagging@mathaddlast
4023   }%
4024   \eql@cell@container
4025   \ifdefined\eql@frame@cmd
4026     \eql@frame@measure
4027     \advance\eql@prevwidth@\eql@frame@margin@
4028     \eql@frame@print
4029   \fi
4030   \kern-\eql@prevwidth@
4031   \unhbox\eql@cellbox@
4032   \hfil
4033   \tabskip\eql@colsep@\relax
4034 \crcr

```

```

4035 \noalign{%
4036 \eq@hook@blockbefore
4037 }%
4038 \eq@hook@blockin
4039 }

4040 \def\eq@box@columns@set{\let\eq@box@open\eq@box@columns@open}

```

11.5 Cases Mode

TODO: describe

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

```

4041 \def\eq@box@cases@lastcell{&}%

4042 \let\eq@box@cases@condtext\eq@false
4043 \let\eq@box@cases@condintro\@empty

4044 \def\eq@box@cases@open{%
4045 \eq@shape@align@disable
4046 \let\eq@box@lastcell\@empty
4047 \everycr{\noalign{%
4048 \eq@verbose@info\eq@verbose@msg@startline@new
4049 }}%
4050 \tabskip\z@skip
4051 \halign\bgroup
4052 \let\eq@box@lastcell\eq@box@cases@lastcell
4053 \global\let\eq@cell@container\@empty
4054 \global\setbox\eq@cellbox@\hbox{%
4055 \eq@strut@cell
4056 \@lign
4057 $\m@th\eq@mathstyle
4058 \eq@hook@colin
4059 ##%
4060 \eq@punct@apply@col
4061 \eq@tagging@mathsave
4062 $%
4063 \eq@tagging@mathaddlast
4064 }%
4065 \eq@cell@container
4066 \unhbox\eq@cellbox@
4067 \hfil
4068 \eq@tagging@alignleft
4069 \tabskip\eq@colsep@\relax
4070 &%
4071 \let\eq@box@lastcell\@empty
4072 \global\let\eq@cell@container\@empty
4073 \setbox\eq@cellbox@\hbox{%
4074 \unhbox\eq@cellbox@
4075 \eq@strut@cell
4076 \@lign
4077 $\m@th\eq@mathstyle
4078 \ifdefined\eq@box@cases@condtext
4079 \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
4080 \eq@box@cases@condintro
4081 ##%
4082 \eq@punct@apply@col
4083 \egroup

```

```

4084         \eql@hook@colout
4085         \eql@tagging@mathsave
4086         $%
4087         \eql@tagging@mathaddlast
4088     }%
4089     \eql@cell@container
4090     \unhbox\eql@cellbox@
4091     \hfil
4092     \eql@tagging@alignleft
4093     \tabskip\z@skip
4094     \crrc
4095     \noalign{%
4096         \eql@hook@blockbefore
4097     }%
4098     \eql@hook@blockin
4099 }

4100 \def\eql@box@cases@set{%
4101     \ifdefined\eql@cases@mathstyle\let\eql@mathstyle\eql@cases@mathstyle\fi
4102     \ifdefined\eql@punct@cases\let\eql@punct@col\eql@punct@cases\fi
4103     \let\eql@box@open\eql@box@cases@open}

```

11.6 Main

```

4104 \let\eql@box@box\center
4105 \let\eql@box@open\@undefined
4106 \let\eql@box@frame\@firstofone
4107 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

4108 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4109     \left#1##1\right#2}}
4110 \def\eql@box@getdim{\setbox\@ne\hbox{\@ne\ht\@ne\ht\z@\dp\@ne\dp\z@}
4111 \def\eql@box@deldim#1{\hbox{$\m@th\hbox{\null\delimiterspace\z@\left#1%
4112     \ifx\eql@box@box\center\center{\box\@ne}\else\box\@ne\fi\right.$}}
4113 \def\eql@box@ldelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4114     \eql@box@getdim\eql@box@deldim#1##1}}
4115 \def\eql@box@rdelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4116     \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!

```

4117 \def\eql@box@close{%
4118     \ifvmode\else
4119         \ifmmode\else\unskip\fi
4120         \global\let\eql@shape@lastrow\eql@true
4121         \eql@punct@apply@block
4122         \ifdefined\eql@box@punct@term
4123             \eql@punct@apply@top
4124         \fi
4125         \eql@box@endline
4126         \eql@box@lastcell
4127         \cr
4128     \fi
4129     \noalign{%
4130         \eql@hook@blockafter

```

```

4131     \global\let\eql@shape@lastrow\eql@false
4132   }%
4133   \eql@tagging@tablesaveinner
4134 \egroup
4135 }

```

\eql@box@vcenter

```

4136 \def\eql@box@vcenter#1{%
4137   \ifmmode
4138     \vcenter{#1}%
4139   \else
4140     $\m@th\vcenter{#1}$%
4141   \fi
4142 }

```

\eql@box@start

```

4143 \let\eql@box@endmath\eql@false
4144 \def\eql@box@start{%
4145   \relax
4146   \ifmmode
4147     \let\eql@box@endmath\eql@false
4148   \else
4149     \let\eql@box@endmath\eql@true
4150     \expandafter$%$
4151   \fi
4152   \eql@beamerbasecolor@avoid
4153   \eql@stack@save@box
4154   \let\eql@frame@cmd\@undefined
4155   \let\eql@layoutleft\eql@false
4156   \eql@row@z@
4157   \eql@totalrows@\@M
4158   \eql@shape@select
4159   \setbox\z@\ifx\eql@box@box\vcenter
4160     \expandafter\vbox
4161   \else
4162     \expandafter\eql@box@box
4163   \fi\bgroup
4164   \let\eqnpunct\eql@punct@setnext
4165   \eql@display@nest
4166   \let\\ \eql@box@cr
4167   \ifdefined\eql@amp@mode
4168     \let\&\eql@box@amp
4169   \fi
4170   \eql@spread@set
4171   \eql@strut@make
4172   \eql@box@open
4173 }

```

\eql@box@end

```

4174 \def\eql@box@end{%
4175   \eql@box@close
4176   \egroup
4177   \eql@box@frame{%
4178     \ifdefined\eql@display@marginleft
4179       \hskip\glueexpr\eql@display@marginleft\relax
4180     \fi

```

```

4181 \ifx\eql@box@box\vcenter
4182 \eql@box@vcenter{\unvbox\z@}%
4183 \else
4184 \box\z@
4185 \fi
4186 \eql@tagging@tableaddinner
4187 \ifdefined\eql@display@marginright
4188 \hskip\glueexpr\eql@display@marginright\relax
4189 \fi
4190 }%
4191 \eql@stack@restore
4192 \ifdefined\eql@box@endmath
4193 \expandafter$%$
4194 \fi
4195 }

```

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

```

4196 \def\eql@box@main{%
4197 \eql@box@start
4198 \eql@scan@body
4199 \eql@box@end
4200 }

```

11.7 Options Processing

TODO: describe

```

4201 \def\eql@box@test@setopt{\let\eql@box@test\eql@box@testopt}
4202 \def\eql@box@test@setall{\let\eql@box@test\eql@box@testall}

```

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

```

4203 \def\eql@box@testopt#1{\eql@testopt@tight{\eql@box@testopt@set{#1}}{}}
4204 \def\eql@box@testopt@set#1[#2]{\eqnaddopt{#2}#1}

4205 \def\eql@box@testall{\eql@parseopt@env\eql@box@testall@parse}
4206 \def\eql@box@testall@parse{%
4207 \ifx\eql@parseopt@token[%
4208 \let\eql@parseopt@next\eql@parseopt@opt
4209 \fi
4210 \ifx\eql@parseopt@token.%
4211 \let\eql@parseopt@next\eql@parseopt@punctpass
4212 \fi
4213 \ifx\eql@parseopt@token,%
4214 \let\eql@parseopt@next\eql@parseopt@punctpass
4215 \fi
4216 \ifx\eql@parseopt@token~%
4217 \let\eql@parseopt@next\eql@parseopt@punctpass
4218 \fi
4219 \ifx\eql@parseopt@token'%
4220 \let\eql@parseopt@next\eql@parseopt@punctopt
4221 \fi
4222 \ifx\eql@parseopt@token!%
4223 \let\eql@parseopt@next\eql@box@parse@punctterm
4224 \fi
4225 \ifx\eql@parseopt@token=%
4226 \let\eql@parseopt@next\eql@parseopt@lines

```



```

4227 \fi
4228 \ifx\eql@parseopt@token|%
4229 \let\eql@parseopt@next\eql@parseopt@columns
4230 \fi
4231 \ifx\eql@parseopt@token<%
4232 \let\eql@parseopt@next\eql@parseopt@ampeq
4233 \fi
4234 \ifx\eql@parseopt@token>%
4235 \let\eql@parseopt@next\eql@parseopt@eqamp
4236 \fi
4237 \ifx\eql@parseopt@token*%
4238 \let\eql@parseopt@next\eql@parseopt@matrix
4239 \fi
4240 \ifx\eql@parseopt@token?%
4241 \let\eql@parseopt@next\eql@parseopt@cases
4242 \fi
4243 \ifx\eql@parseopt@token\eql@atxi
4244 \let\eql@parseopt@next\eql@parseopt@shape
4245 \fi
4246 \ifx\eql@parseopt@token\eql@atxii
4247 \let\eql@parseopt@next\eql@parseopt@shape
4248 \fi
4249 }
4250 \def\eql@box@parse@punctterm#1{\eqnaddopt{punctterm}\eql@parseopt@peek}

```

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

```

4251 \def\eql@box@end@testall{\eql@parseopt@env\eql@box@end@testall@parse}
4252 \def\eql@box@end@testall@parse{%
4253 \ifx\eql@parseopt@token.%
4254 \let\eql@parseopt@next\eql@parseopt@punctpass
4255 \fi
4256 \ifx\eql@parseopt@token,%
4257 \let\eql@parseopt@next\eql@parseopt@punctpass
4258 \fi
4259 \ifx\eql@parseopt@token~%
4260 \let\eql@parseopt@next\eql@parseopt@punctpass
4261 \fi
4262 \ifx\eql@parseopt@token'%
4263 \let\eql@parseopt@next\eql@parseopt@punctblock
4264 \fi
4265 \ifx\eql@parseopt@token!%
4266 \let\eql@parseopt@next\eql@box@end@parse@punctterm
4267 \fi
4268 \ifx\eql@parseopt@token$%$
4269 \let\eql@parseopt@next\eql@parseopt@end
4270 \fi
4271 }
4272 \def\eql@box@end@parse@punctterm#1{%
4273 \let\eql@box@punct@term\eql@true\eql@parseopt@peek}

```

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

```

4274 \def\eql@box@processopt{%
4275 \let\eql@box@frame\@firstofone
4276 \let\eql@display@marginleft\@undefined
4277 \let\eql@display@marginright\@undefined
4278 \let\eql@box@punct@term\eql@false
4279 \let\eql@punct@block\@undefined

```

```

4280 \eql@nextopt@process{box}%
4281 \let\eql@punct@next\@undefined
4282 \eql@colsep@\glueexpr\eql@box@colsep\relax
4283 \ifdefined\eql@paddingleft@val
4284 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
4285 \else
4286 \eql@paddingleft@0\z@
4287 \fi
4288 \ifdefined\eql@paddingright@val
4289 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
4290 \else
4291 \eql@paddingright@0\z@
4292 \fi
4293 \eql@indent@\glueexpr\eql@indent@val\relax
4294 }

```

11.8 Environment

`equationsbox` (*env.*)

```

4295 \newenvironment{equationsbox}{%
4296 \eql@verbose@info\eql@verbose@msg@enterenv
4297 \ifdefined\eql@box@env@modifier
4298 \eql@box@test@setall
4299 \else
4300 \eql@box@nomodifier
4301 \fi
4302 \eql@ampprotect\eql@box@test\eql@box@env@start
4303 }{%
4304 \ifdefined\eql@box@doscan\else
4305 \expandafter\eql@box@end
4306 \fi
4307 \eql@verbose@info\eql@verbose@msg@leaveenv
4308 }

```

`\eql@box@env@start`

```

4309 \def\eql@box@env@start{%
4310 \eql@box@processopt
4311 \ifdefined\eql@box@doscan
4312 \eql@box@call@set
4313 \expandafter\eql@scan@env
4314 \else
4315 \expandafter\eql@box@start
4316 \fi
4317 }

```

`\eql@box@call`

```

4318 \def\eql@box@call{\eql@box@main\eql@scan@end}
4319 \def\eql@box@call@test{\eql@ampprotect\eql@box@end@testall\eql@box@call}
4320 \def\eql@box@call@set{%
4321 \ifdefined\eql@box@end@modifier
4322 \let\eql@scan@call\eql@box@call@test
4323 \else
4324 \let\eql@scan@call\eql@box@call
4325 \fi
4326 }

```

`\eql@box@ang@open`

```
4327 \newenvironment{equationsbox@ang}{-}{-}
4328 \def\eql@box@ang@open{%
4329   \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4330   \begin{equationsbox@ang}%
4331   \eql@verbose@info\eql@verbose@msg@enterenv
4332   \let\>\eql@box@ang@close
4333   \ifdefined\eql@box@ang@modifier
4334     \eql@box@test@setall
4335   \else
4336     \eql@box@nomodifier
4337   \fi
4338   \eql@ampprotect\eql@box@test\eql@box@ang@start
4339 }
```

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

```
4340 \def\eql@box@ang@start{%
4341   \eql@box@processopt
4342   \ifdefined\eql@box@doscan
4343     \eql@box@call@set
4344     \expandafter\eql@scan@ang
4345   \else
4346     \expandafter\eql@box@start
4347   \fi
4348 }
```

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

```
4349 \def\eql@box@ang@close{%
4350   \ifdefined\eql@box@doscan
4351     \let\eql@box@end@modifier\eql@false
4352   \fi
4353   \ifdefined\eql@box@end@modifier
4354     \expandafter\eql@ampprotect\expandafter\eql@box@end@testall
4355   \fi
4356   \eql@box@ang@end
4357 }
```

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

```
4358 \def\eql@box@ang@end{%
4359   \ifdefined\eql@box@doscan\else
4360     \expandafter\eql@box@end
4361   \fi
4362   \eql@verbose@info\eql@verbose@msg@leaveenv
4363   \end{equationsbox@ang}%
4364   \ignorespaces
4365 }
```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

4366 \def\eql@single@start@native{%
4367   \eql@display@init
4368   \eql@display@print
4369   \let\raisetag\eql@raisetag@default
4370   \eql@shape@align@disable
4371   \eql@hook@eqin
4372 }%

```

TODO: describe

```

4373 \def\eql@single@end@native{%
4374   \eql@nextopt@apply{equations}%
4375   \eql@tags@container
4376   \eql@numbering@single@eval
4377   \if@eqnsw
4378     \ifdefined\eql@tagsleft
4379       \leqno
4380     \else
4381       \eqno
4382     \fi
4383     \eql@composetag@print
4384   \fi
4385   \eql@interline@container
4386   \advance\eql@belowspace@\eql@vspaceskip@
4387   \eql@display@container
4388   \eql@display@penalty
4389   \eql@display@vspace@native
4390 }%

```

12.2 Print

```

4391 \def\eql@single@start@print{%
4392   \eql@display@init
4393   \eql@display@print
4394   \eql@shape@align@enable
4395   \eql@totalrows@\@ne
4396   \eql@row@\@ne
4397   \eql@arrange@init
4398   \global\let\eql@cell@container\@empty
4399   \prevgraf\numexpr\prevgraf+\@ne\relax
4400   \setbox\eql@cellbox@\hbox\bgroup
4401     \eql@restore@hfuzz
4402     \eql@strut@cell
4403     $\m@th\eql@mathstyle%$
4404     \eql@hook@eqin
4405 }
4406 \def\eql@single@end@print{%
4407   \global\let\eql@nextopt@tmp\eql@nextopt
4408   \eql@tagging@mathsave
4409   $%$
4410   \hfil
4411   \kern\z@
4412   \egroup
4413   \prevgraf\numexpr\prevgraf-\@ne\relax
4414   \let\eql@nextopt\eql@nextopt@tmp
4415   \eql@nextopt@apply{equations}%
4416   \eql@shape@eval
4417   \eql@cell@container

```

```

4418 \ifdefined\eql@frame@cmd
4419 \eql@frame@adjust
4420 \fi

4421 \eql@cellwidth@\wd\eql@cellbox@
4422 \eql@line@height@\ht\eql@cellbox@
4423 \eql@line@depth@\dp\eql@cellbox@
4424 \eql@totalwidth@\eql@cellwidth@
4425 \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
4426 \eql@topheight@\eql@line@height@
4427 \eql@bottomdepth@\eql@line@depth@

4428 \eql@tags@container
4429 \eql@numbering@single@eval
4430 \if@eqnsw
4431 \eql@tagbox@make\eql@composetag@print
4432 \eql@tagrows@\@ne
4433 \ifdefined\eql@tagpos@reserve\else
4434 \eql@tagwidth@\z@
4435 \fi
4436 \eql@tagheight@block@\ht\eql@tagbox@
4437 \eql@tagdepth@block@\dp\eql@tagbox@
4438 \else
4439 \eql@numbering@warnunused
4440 \eql@tagwidth@\z@
4441 \eql@tagrows@\z@
4442 \fi
4443 \eql@tagwidth@max@\eql@tagwidth@
4444 \eql@tagpos@single@eval
4445 \eql@tagpos@print@line@eval

4446 \eql@intercolumns@\z@
4447 \eql@adjust@calc@lines

4448 \eql@display@halign@init{}%
4449 \halign{##\crr
4450 \noalign{\eql@display@halign@start}%
4451 \eql@arrange@print@line
4452 \cr
4453 \noalign{\eql@display@halign@end}%
4454 \eql@tagging@tablesavelines
4455 }%
4456 \eql@tagpos@print@line@end
4457 \eql@display@close
4458 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```

4459 \def\eql@lines@measure@line@begin{%
4460 \eql@verbose@info\eql@verbose@msg@startline@number
4461 \eql@numbering@measure@line@begin
4462 \eql@hook@linein
4463 }

```

TODO: describe

```
4464 \def\eql@lines@measure@line@end{%
4465   \eql@punct@apply@line
4466   \eql@hook@lineout
4467 }
```

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

```
4468 \def\eql@lines@measure@cell{%
4469   \ifdefined\eql@frame@cmd
4470     \ifcase\eql@shape@pos@
4471       \eql@frame@measure
4472       \advance\eql@shape@amount@-\eql@frame@margin@
4473     \or\or
4474       \eql@frame@measure
4475       \advance\eql@shape@amount@+\eql@frame@margin@
4476     \fi
4477     \eql@frame@print
4478   \fi
4479   \eql@cellwidth@\wd\eql@cellbox@
4480   \eql@line@height@\ht\eql@cellbox@
4481   \eql@line@depth@\dp\eql@cellbox@
4482   \eql@dimensions@startrow
4483   \eql@dimensions@savecell
4484   \kern\eql@cellwidth@
4485 }
```

`\eql@lines@measure`

```
4486 \def\eql@lines@measure{%
4487   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
4488   \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
4489   \ifdefined\eql@amp@mode
4490     \let\&\eql@break@amp
4491   \fi
4492   \eql@totalrows@\@M
4493   \eql@shape@select

4494   \setbox\z@\vbox{\measuring@true\halign{%
4495     \global\let\eql@cell@container\@empty
4496     \setbox\eql@cellbox@\hbox{%
4497       \eql@strut@cell
4498       \@lign
4499       $\m@th\eql@mathstyle
4500       \eql@hook@colin
4501       ##%
4502       \eql@punct@apply@col
4503       \eql@hook@colout
4504       $%
4505     }%
4506     \ifdefined\eql@shape@lastrow
4507       \eql@totalrows@\eql@row@
4508     \fi
4509     \eql@shape@eval
4510     \eql@cell@container
4511     \eql@lines@measure@cell
4512     \eql@measure@tag
4513     \eql@measure@endrow
```

```

4514 \crrr

4515 \noalign{%
4516   \global\let\eq\@shape@lastrow\eq\@false
4517   \eq\@hook@blockbefore
4518 }%
4519 \eq\@hook@blockin
4520 \eq\@scan@body
4521 \ifvmode\else
4522   \global\let\eq\@shape@lastrow\eq\@true
4523   \eq\@punct@apply@block
4524   \eq\@hook@blockout
4525   \eq\@multi@endline
4526   \cr
4527 \fi
4528 \omit
4529 \cr
4530 \noalign{%
4531   \eq\@hook@blockafter
4532   \global\let\eq\@shape@lastrow\eq\@false
4533 }%
4534 }}%

4535 \eq\@measure@close

4536 \setbox\z@\vbox{%
4537   \unvbox\z@
4538   \unpenalty
4539   \global\setbox\@ne\lastbox
4540 }%
4541 \eq\@totalwidth@\wd\@ne

4542 \eq\@verbose@infoarg\eq\@verbose@msg@leave\eq\@lines@measure
4543 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

4544 \def\eq\@lines@adjust{%
4545   \eq\@tagpos@adjust@eval
4546   \eq\@adjust@calc@lines
4547   \eq\@numbering@best@eval
4548 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```

4549 \def\eq\@lines@print@line@begin{%
4550   \eq\@verbose@info\eq\@verbose@msg@startline@number
4551   \eq\@numbering@print@line@begin
4552   \eq\@hook@linein
4553 }

```

TODO: describe

```
4554 \def\eql@lines@print@line@end{%
4555   \eql@punct@apply@line
4556   \eql@hook@lineout
4557 }
```

TODO: describe

```
4558 \def\eql@lines@print@line@adjust{%
4559   \ifdefined\eql@frame@cmd
4560     \ifcase\eql@shape@pos@
4561       \eql@frame@measure
4562       \advance\eql@shape@amount@-\eql@frame@margin@
4563     \or\or
4564       \eql@frame@measure
4565       \advance\eql@shape@amount@+\eql@frame@margin@
4566     \fi
4567     \eql@frame@adjust
4568   \fi
4569   \eql@cellwidth@\wd\eql@cellbox@
4570   \eql@line@height@\ht\eql@cellbox@
4571   \eql@line@depth@\dp\eql@cellbox@
4572   \eql@numbering@print@line@eval
4573   \if@eqnsw
4574     \eql@tagbox@make\eql@composetag@print
4575   \fi
4576   \eql@tagpos@print@line@eval
4577   \eql@arrange@print@line
4578   \eql@tagpos@print@line@end
4579 }
```

TODO: describe

```
4580 \def\eql@lines@print{%
4581   \eql@verbose@infoarg\eql@verbose@msg@center\eql@lines@print
4582   \eql@arrange@init
4583   \eql@display@halign@init\eql@lines@print@line@begin
4584   \eql@multi@cr@let\eql@lines@print@line@end
4585   \ifdefined\eql@amp@mode
4586     \let\&\eql@break@amp
4587   \fi
4588   \tabskip\z@skip

4589   \halign{%
4590     \global\let\eql@cell@container\@empty
4591     \setbox\eql@cellbox@\hbox{%
4592       \eql@restore@hfuzz
4593       \eql@strut@cell
4594       \@lign
4595       $\m@th\eql@mathstyle
4596       \eql@hook@colin
4597       ##%
4598       \eql@punct@apply@col
4599       \eql@hook@colout
4600       \eql@tagging@mathsave
4601       $%
4602       \hfil
4603       \kern\z@
4604     }%
```



```

4605     \eql@shape@eval
4606     \eql@cell@container
4607     \eql@lines@print@line@adjust
4608     \crrr

4609     \noalign{%
4610         \eql@display@halign@start
4611         \eql@numbering@print@block@begin
4612         \eql@hook@blockbefore
4613     }%
4614     \eql@hook@blockin
4615     \eql@scan@body
4616     \ifvmode\else
4617         \relax
4618         \eql@punct@apply@block
4619         \eql@hook@blockout
4620         \eql@multi@endline
4621         \cr
4622     \fi
4623     \noalign{%
4624         \eql@hook@blockafter
4625         \eql@display@halign@end
4626         \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@print
4627     }%
4628     \eql@tagging@tablesavelines
4629 }%
4630 }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```

\eql@columns@add@amp
@columns@completerow
4631 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
4632 \def\eql@columns@completerow{%
4633     \count@ \numexpr\eql@totalcolumns@+\@ne-\eql@column@ \relax
4634     \edef\eql@tmp{%
4635         \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
4636     \eql@tmp
4637 }

4638 \def\eql@columns@overfull{%
4639     \dimen@ \eql@line@width@
4640     \advance\dimen@-\hfuzz
4641     \ifdim\dimen@>\displaywidth
4642         \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
4643         \wd\z@\z@
4644         \ht\z@\eql@line@height@
4645         \dp\z@\eql@line@depth@
4646         \box\z@
4647     \fi
4648 }

```

14.2 Column Breaks

TODO: describe

TODO: describe

```
4649 \let\eql@amp@org\&

4650 \def\eql@columns@amp{%
4651   \eql@ampprotecttwo\eql@columns@amp@testescape\eql@amp@org
4652   \eql@columns@amp@process}
4653 \def\eql@columns@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
4654   \relax
4655   \let\eql@punct@term\eql@false
4656   \let\eql@class@rel@composed\@empty
4657   \eql@columns@amp@test{#2}}}
```

TODO: describe

```
4658 \def\eql@columns@amp@test@setopt{%
4659   \let\eql@columns@amp@test\eql@columns@amp@testopt}
4660 \def\eql@columns@amp@test@setall{%
4661   \let\eql@columns@amp@test\eql@columns@amp@testall}
```

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

```
4662 \let\eql@columns@amp@testopt\@empty
```

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

ns@amp@testall@parse

```
4663 \def\eql@columns@amp@testall{\eql@parseopt@cr\eql@columns@amp@testall@parse}
4664 \def\eql@columns@amp@testall@parse{%
4665   \ifx\eql@parseopt@token.%
4666     \let\eql@parseopt@next\eql@parseopt@punctpass
4667   \fi
4668   \ifx\eql@parseopt@token,%
4669     \let\eql@parseopt@next\eql@parseopt@punctpass
4670   \fi
4671   \ifx\eql@parseopt@token~%
4672     \let\eql@parseopt@next\eql@parseopt@punctpass
4673   \fi
4674   \ifx\eql@parseopt@token'
4675     \let\eql@parseopt@next\eql@parseopt@punctnext
4676   \fi
4677   \ifx\eql@parseopt@token!%
4678     \let\eql@parseopt@next\eql@parseopt@punctterm
4679   \fi
4680   \ifx\eql@parseopt@token=%
4681     \let\eql@parseopt@next\eql@parseopt@relsymp
4682   \fi
4683   \ifx\eql@parseopt@token;%
4684     \let\eql@parseopt@next\eql@parseopt@relcont
4685   \fi
4686   \ifx\eql@parseopt@token:%
4687     \let\eql@parseopt@next\eql@parseopt@relstart
4688   \fi
4689   \ifx\eql@parseopt@token|%
4690     \let\eql@parseopt@next\eql@parseopt@relord
4691   \fi
4692   \ifx\eql@parseopt@token?%
```

```

4693 \let\eql@parseopt@next\eql@columns@amp@parse@rel
4694 \fi
4695 \ifx\eql@parseopt@token&%
4696 \let\eql@parseopt@next\eql@parseopt@end
4697 \fi
4698 }
4699 \def\eql@columns@amp@parse@rel#1#2{%
4700 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
4701 \eql@parseopt@end}

```

@columns@amp@process

```

4702 \def\eql@columns@amp@process{%
4703 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
4704 \edef\eql@tmp{%
4705 \ifx\eql@class@rel@composed\@empty
4706 \ifodd\eql@column@&\noexpand\eql@punct@next@clear\fi&%
4707 \else
4708 \ifodd\eql@column@\else&\fi&%
4709 \unexpanded\expandafter{\eql@class@rel@composed}%
4710 \fi
4711 }%
4712 \eql@tmp
4713 }

```

14.3 Transpose

TODO: describe

TODO: adjust to \&?!

TODO: describe

```

4714 \let\eql@transpose@active\eql@false
4715 \def\eql@transpose@end{\eql@transpose@end}
4716 \def\eql@transpose@skip{\&\eql@punct@next@clear}
4717 \def\eql@transpose@complete{%
4718 \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

TODO: describe

```

4719 \def\eql@transpose{%
4720 \eql@totalcolumns@z@
4721 \eql@totalrows@z@
4722 \expandafter\eql@transpose@scan@col\the\eql@scan@reg@\&\eql@transpose@end\&%
4723 \eql@scan@reg@{}%
4724 \eql@row@z@
4725 \eql@transpose@output@row
4726 }

```

TODO: describe

```

4727 \def\eql@transpose@save@col#1{%
4728 \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{}%
4729 \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

TODO: describe

```

4730 \def\eql@transpose@scan@col#1\&{%
4731 \def\eql@tmpa{#1}%
4732 \ifx\eql@tmpa\eql@transpose@end\else

```

```

4733 \advance\eql@totalcolumns@\@ne
4734 \eql@row@\z@
4735 \let\eql@transpose@data@col\@empty
4736 \eql@transpose@scan@row#1\\eql@transpose@end\\%
4737 \ifnum\eql@row@>\eql@totalrows@
4738 \eql@totalrows@\eql@row@
4739 \fi
4740 \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
4741 \expandafter\eql@transpose@scan@col
4742 \fi
4743 }

```

TODO: describe

```

4744 \def\eql@transpose@append@row#1{%
4745 \advance\eql@row@\@ne
4746 \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}

```

TODO: describe

```

4747 \def\eql@transpose@scan@row#1\\{%
4748 \def\eql@tmpa{#1}%
4749 \ifx\eql@tmpa\eql@transpose@end\else
4750 \ifx\eql@transpose@active+
4751 \eql@transpose@scan@cell#1&\eql@transpose@end&%
4752 \else
4753 \eql@transpose@append@row{#1}%
4754 \fi
4755 \expandafter\eql@transpose@scan@row
4756 \fi
4757 }

```

TODO: describe

```

4758 \def\eql@transpose@scan@cell#1&#2&{%
4759 \def\eql@tmpa{#2}%
4760 \ifx\eql@tmpa\eql@transpose@end
4761 \eql@transpose@append@row{#1}%
4762 \else
4763 \eql@transpose@append@row{#1&#2}%
4764 \expandafter\eql@transpose@scan@cell@next
4765 \fi
4766 }

```

TODO: describe

```

4767 \def\eql@transpose@scan@cell@next#1&{%
4768 \def\eql@tmpa{#1}%
4769 \ifx\eql@tmpa\eql@transpose@end\else
4770 \eql@transpose@append@row{&#1}%
4771 \expandafter\eql@transpose@scan@cell@next
4772 \fi
4773 }

```

TODO: describe

```

4774 \def\eql@transpose@output@row{%
4775 \ifnum\eql@row@<\eql@totalrows@
4776 \advance\eql@row@\@ne
4777 \eql@column@\z@
4778 \eql@transpose@output@col
4779 \ifnum\eql@row@<\eql@totalrows@

```

```

4780     \eql@scan@addto\\%
4781     \fi
4782     \expandafter\eql@transpose@output@row
4783     \fi
4784 }

```

TODO: describe

```

4785 \def\eql@transpose@output@col{%
4786   \ifnum\eql@column@<\eql@totalcolumns@
4787     \advance\eql@column@ \@ne
4788     \csname eql@transpose@data@col@\the\eql@column@\endcsname
4789     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4790     \ifnum\eql@column@<\eql@totalcolumns@
4791       \eql@scan@addto{\eql@transpose@complete}%
4792     \fi
4793     \expandafter\eql@transpose@output@col
4794   \fi
4795 }

```

14.4 Measure

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

s@measure@line@begin

```

4796 \def\eql@columns@measure@line@begin{%
4797   \eql@verbose@info\eql@verbose@msg@startline@number
4798   \global\eql@column@\z@
4799   \global\eql@line@height@\z@
4800   \global\eql@line@depth@\z@
4801   \eql@numbering@measure@line@begin
4802   \eql@hook@linein
4803 }

4804 \def\eql@columns@measure@cell{%
4805   \eql@cellwidth@\wd\eql@cellbox@
4806   \ifdefined\eql@frame@cmd
4807     \eql@frame@measure
4808     \advance\eql@cellwidth@\eql@frame@margin@
4809   \fi
4810   \ifdim\ht\eql@cellbox@>\eql@line@height@
4811     \global\eql@line@height@\ht\eql@cellbox@
4812   \fi
4813   \ifdim\dp\eql@cellbox@>\eql@line@depth@
4814     \global\eql@line@depth@\dp\eql@cellbox@
4815   \fi
4816   \ifnum\eql@column@=\@ne
4817     \eql@dimensions@startrow
4818   \fi
4819   \ifodd\eql@column@
4820     \eql@shape@pos@\tw@
4821   \else
4822     \eql@shape@pos@\z@
4823   \fi
4824   \eql@shape@amount@\z@
4825   \eql@dimensions@savecell
4826   \ifodd\eql@column@\else
4827     \eql@dimensions@savessep

```

```

4828 \fi
4829 \kern\eql@cellwidth@
4830 }

```

ms@measure@line@end

```

4831 \def\eql@columns@measure@line@end{%
4832 \eql@punct@apply@line
4833 \eql@hook@lineout
4834 &\omit
4835 \ifnum\eql@column@>\eql@totalcolumns@
4836 \global\eql@totalcolumns@\eql@column@
4837 \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

```

4838 \ifdefined\eql@frame@cmd
4839 \advance\eql@column@\@ne
4840 \wd\eql@cellbox@z@
4841 \eql@columns@measure@cell
4842 \fi
4843 \eql@measure@tag
4844 \eql@measure@endrow
4845 }

```

\eql@columns@measure

```

4846 \def\eql@columns@measure{%
4847 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@measure
4848 \eql@totalcolumns@z@
4849 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end
4850 \ifdefined\eql@amp@mode
4851 \let&\eql@columns@amp
4852 \fi

4853 \setboxz@\vbox{\measuring@true\halign{%
4854 &%
4855 \global\advance\eql@column@\@ne
4856 \global\let\eql@cell@container\@empty
4857 \global\setbox\eql@cellbox@\hbox{%
4858 \eql@strut@cell
4859 \@lign
4860 $\m@th\eql@mathstyle
4861 \eql@hook@colin
4862 ##%
4863 \eql@punct@apply@next
4864 \eql@class@innerleft
4865 \eql@hook@innerleft
4866 $%
4867 }%
4868 \eql@cell@container
4869 \hfil
4870 \eql@columns@measure@cell
4871 \global\let\eql@frame@prevcmd\eql@frame@cmd
4872 &%
4873 \eql@prevwidth@\wd\eql@cellbox@
4874 \let\eql@frame@cmd\eql@frame@prevcmd

```

```

4875     \global\advance\eql@column@\@ne
4876     \global\let\eql@cell@container\@empty
4877     \setbox\eql@cellbox@\hbox{%
4878         \eql@strut@cell
4879         \@lign
4880         $\m@th\eql@mathstyle
4881         \eql@hook@innerright
4882         \eql@class@innerright@sel
4883         ##%
4884         \eql@punct@apply@col
4885         \eql@hook@colout
4886         $%
4887     }%
4888     \eql@cell@container
4889     \eql@columns@measure@cell
4890     \hfil
4891     \crrr

4892     \noalign{%
4893         \eql@hook@blockbefore
4894     }%
4895     \eql@hook@blockin
4896     \eql@scan@body

4897     \ifvmode\else
4898         \eql@punct@apply@block
4899         \eql@hook@blockout
4900         \eql@multi@endline
4901         \cr
4902     \fi
4903     \noalign{%
4904         \eql@hook@blockafter
4905     }%

```

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

```

4906     \omit
4907     \eql@column@\@ne
4908     \eql@columns@completerow
4909     \cr
4910     }}%

4911     \eql@measure@close

4912     \setbox\z@\vbox{%
4913         \unvbox\z@
4914         \unpenalty
4915         \global\setbox\@ne\lastbox
4916     }%
4917     \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4918     \let\eql@colwidth@tab\@empty
4919     \loop
4920         \setbox\@ne\hbox{%
4921             \unhbox\@ne
4922             \unskip
4923             \global\setbox\thr@\@ne\lastbox
4924         }%

```

```

4925 \ifhbox\thr@@
4926 \eql@colwidth@save{\wd\thr@@}%
4927 \repeat

4928 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@measure
4929 }

```

14.5 Columns Placement

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

```

4930 \def\eql@columns@adjust{%
4931 \ifodd\eql@totalcolumns@
4932 \advance\eql@totalcolumns@\@ne
4933 \fi
4934 \eql@tagpos@adjust@eval
4935 \eql@adjust@calc@columns
4936 }

```

14.6 Print

TODO: describe

ms@print@line@begin

```

4937 \def\eql@columns@print@line@begin{%
4938 \eql@verbose@info\eql@verbose@msg@startline@number
4939 \global\eql@column@\z@
4940 \global\eql@line@pos@\eql@marginleft@
4941 \global\eql@line@width@\z@
4942 \global\eql@line@avail@\eql@totalwidth@
4943 \global\eql@line@height@\z@
4944 \global\eql@line@depth@\z@
4945 \eql@numbering@print@line@begin
4946 \eql@hook@linein
4947 }

```

l@columns@print@cell

```

4948 \def\eql@columns@print@cell{%
4949 \eql@cellwidth@\wd\eql@cellbox@
4950 \ifodd\eql@column@
4951 \ifdefined\eql@frame@cmd
4952 \eql@frame@measure
4953 \advance\eql@cellwidth@\eql@frame@margin@
4954 \fi
4955 \dimen@\z@
4956 \else
4957 \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```

4958 \ifdefined\eql@frame@cmd
4959 \eql@frame@measure
4960 \advance\eql@cellwidth@\eql@frame@margin@
4961 \advance\eql@prevwidth@\eql@frame@margin@
4962 \eql@frame@print

```


4963 \fi

update height and depth

```
4964     \ifdim\ht\eql@cellbox@>\eql@line@height@
4965         \global\eql@line@height@\ht\eql@cellbox@
4966     \fi
4967     \ifdim\dp\eql@cellbox@>\eql@line@depth@
4968         \global\eql@line@depth@\dp\eql@cellbox@
4969     \fi
```

print box

```
4970     \kern-\eql@prevwidth@
4971     \unhbox\eql@cellbox@
4972     \dimen@-\eql@cellwidth@
4973     \fi
```

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

```
4974     \advance\dimen@\eql@colwidth@get\eql@column@\relax
4975     \kern\dimen@
```

update available and used space

```
4976     \dimen@\eql@colwidth@get\eql@column@\relax
4977     \ifdim\eql@cellwidth@>\z@
4978         \ifdim\eql@line@width@=\z@
4979             \eql@line@avail@\eql@line@pos@
4980             \ifodd\eql@column@
4981                 \advance\eql@line@avail@\dimen@
4982                 \advance\eql@line@avail@-\eql@cellwidth@
4983             \fi
4984             \global\eql@line@avail@\eql@line@avail@
4985         \fi
4986         \eql@line@width@\eql@line@pos@
4987         \ifodd\eql@column@
4988             \advance\eql@line@width@\dimen@
4989         \else
4990             \advance\eql@line@width@\eql@cellwidth@
4991         \fi
4992         \global\eql@line@width@\eql@line@width@
4993     \fi
4994     \advance\eql@line@pos@\dimen@
4995     \ifodd\eql@column@\else
4996         \advance\eql@line@pos@\eql@colsep@
4997     \fi
4998     \global\eql@line@pos@\eql@line@pos@
4999 }
```

```
5000 \def\eql@columns@print@trailright{%
5001     &\omit
5002     \eql@prevwidth@\wd\eql@cellbox@
5003     \let\eql@frame@cmd\eql@frame@prevcmd
5004     \global\advance\eql@column@\@ne
5005     \eql@columns@print@cell
5006 }
```

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

```

5007 \def\eql@columns@print@line@end{%
5008   \eql@punct@apply@line
5009   \eql@hook@lineout
5010   \ifodd\eql@column@
5011     \expandafter\eql@columns@print@trailright
5012   \fi
5013   \eql@columns@completerow
5014   \eql@columns@print@tag
5015 }

```

ql@columns@print@tag

```

5016 \def\eql@columns@print@tag{%
5017   \kern-\dimexpr\eql@totalwidth+ \eql@colsep@ \relax

```

determine first line available space

```

5018   \eql@display@firstavail@set\eql@line@avail@
5019   \eql@columns@overfull
5020   \eql@numbering@print@line@eval
5021   \if@eqnsw
5022     \eql@tagbox@make\eql@composetag@print
5023   \fi
5024   \eql@tagpos@print@line@eval
5025   \eql@tagbox@print@cell
5026   \eql@tagpos@print@line@end
5027 }

```

\eql@columns@print

```

5028 \def\eql@columns@print{%
5029   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
5030   \eql@shape@align@disable
5031   \eql@display@halign@init\eql@columns@print@line@begin
5032   \eql@multi@cr@let\eql@columns@print@line@end
5033   \ifdefined\eql@amp@mode
5034     \let\&\eql@columns@amp
5035   \fi
5036   \tabskip\eql@marginleft@

5037   \halign{%
5038     &%
5039     \global\advance\eql@column@\@ne
5040     \global\let\eql@cell@container\@empty
5041     \global\setbox\eql@cellbox@\hbox{%
5042       \eql@strut@cell
5043       \@lign
5044       $\m@th\eql@mathstyle
5045       \eql@hook@colin
5046       ##%
5047       \eql@punct@apply@next
5048       \eql@class@innerleft
5049       \eql@hook@innerleft
5050       \eql@tagging@mathsave
5051       $%
5052       \eql@tagging@mathaddlast
5053     }%
5054     \eql@cell@container
5055     \hfil
5056     \eql@columns@print@cell

```

```

5057 \global\let\eql@frame@prevcmd\eql@frame@cmd
5058 \tabskip\z@skip
5059 &%
5060 \eql@prevwidth@\wd\eql@cellbox@
5061 \let\eql@frame@cmd\eql@frame@prevcmd
5062 \global\advance\eql@column@\@ne
5063 \global\let\eql@cell@container\@empty
5064 \setbox\eql@cellbox@\hbox{%
5065 \unhbox\eql@cellbox@
5066 \eql@strut@cell
5067 \@lign
5068 $\m@th\eql@mathstyle
5069 \eql@hook@innerright
5070 \eql@class@innerright@sel
5071 ##%
5072 \eql@punct@apply@col
5073 \eql@hook@colout
5074 \eql@tagging@mathsave
5075 $%
5076 \eql@tagging@mathaddlast
5077 }%
5078 \eql@cell@container
5079 \eql@columns@print@cell
5080 \hfil
5081 \tabskip\eql@colsep@\relax
5082 \crrcr

5083 \noalign{%
5084 \eql@display@halign@start
5085 \eql@numbering@print@block@begin
5086 \eql@hook@blockbefore
5087 }%
5088 \eql@hook@blockin
5089 \eql@scan@body
5090 \ifvmode\else
5091 \relax
5092 \eql@punct@apply@block
5093 \eql@hook@blockout
5094 \eql@multi@endline
5095 \cr
5096 \fi
5097 \noalign{%
5098 \eql@hook@blockafter
5099 \eql@display@halign@end
5100 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
5101 }%
5102 \eql@tagging@tablesalign
5103 }%
5104 }

```

15 Interface

15.1 Options Processing

TODO: describe

```

5105 \def\eql@equations@test@setopt{\let\eql@equations@test\eql@equations@testopt}

```

```

5106 \def\eqlequations@test@setall{\let\eqlequations@test\eqlequations@testall}

\eqlequations@testopt TODO: describe

5107 \def\eqlequations@testopt#1{%
5108   \eqlequations@test@tight{\eqlequations@testopt@set{#1}}{}}
5109 \def\eqlequations@testopt@set#1[#2]{\eqnaddopt{#2}#1}

\eqlequations@testall The macro sequence started by \eqlequations@testall scans for optional arguments to
the equation environments and appends them to the argument list using \eqnaddopt. All
arguments are scanned such that any spaces stop the scanning and such that any
alignment markers ‘&’ cannot interfere: TODO: update

5110 \def\eqlequations@testall{\eqlequations@testall@parse}
5111 \def\eqlequations@testall@parse{%
5112   \ifx\eqlequations@testopt@token*%
5113     \let\eqlequations@testopt@next\eqlequations@testopt@nonumber
5114   \fi
5115   \ifx\eqlequations@testopt@token!%
5116     \let\eqlequations@testopt@next\eqlequations@testopt@donumber
5117   \fi
5118   \ifx\eqlequations@testopt@token/%
5119     \let\eqlequations@testopt@next\eqlequations@testopt@transpose
5120   \fi
5121   \ifx\eqlequations@testopt@token[%]
5122     \let\eqlequations@testopt@next\eqlequations@testopt@opt
5123   \fi
5124   \ifx\eqlequations@testopt@token\eqlequations@testopt@atxi
5125     \let\eqlequations@testopt@next\eqlequations@testopt@label
5126   \fi
5127   \ifx\eqlequations@testopt@token\eqlequations@testopt@atxii
5128     \let\eqlequations@testopt@next\eqlequations@testopt@label
5129   \fi
5130   \ifx\eqlequations@testopt@token.%
5131     \let\eqlequations@testopt@next\eqlequations@testopt@punctpass
5132   \fi
5133   \ifx\eqlequations@testopt@token,%
5134     \let\eqlequations@testopt@next\eqlequations@testopt@punctpass
5135   \fi
5136   \ifx\eqlequations@testopt@token~%
5137     \let\eqlequations@testopt@next\eqlequations@testopt@punctpass
5138   \fi
5139   \ifx\eqlequations@testopt@token'%
5140     \let\eqlequations@testopt@next\eqlequations@testopt@punctopt
5141   \fi
5142   \ifx\eqlequations@testopt@token-%
5143     \let\eqlequations@testopt@next\eqlequations@testopt@single
5144   \fi
5145   \ifx\eqlequations@testopt@token=%
5146     \let\eqlequations@testopt@next\eqlequations@testopt@lines
5147   \fi
5148   \ifx\eqlequations@testopt@token|%
5149     \let\eqlequations@testopt@next\eqlequations@testopt@columns
5150   \fi
5151   \ifx\eqlequations@testopt@token<%
5152     \let\eqlequations@testopt@next\eqlequations@testopt@ampeq
5153   \fi
5154   \ifx\eqlequations@testopt@token>%
5155     \let\eqlequations@testopt@next\eqlequations@testopt@eqamp

```

```

5156 \fi
5157 \ifx\eql@parseopt@token\label
5158   \let\eql@parseopt@next\eql@parseopt@end
5159 \fi
5160 \ifx\eql@parseopt@token\begin
5161   \let\eql@parseopt@next\eql@parseopt@end
5162 \fi
5163 }

```

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

```

5164 \def\eql@equations@end@testall{%
5165   \eql@parseopt@env\eql@equations@end@testall@parse}
5166 \def\eql@equations@end@testall@parse{%
5167   \ifx\eql@parseopt@token[%
5168     \let\eql@parseopt@next\eql@parseopt@opt
5169   \fi
5170   \ifx\eql@parseopt@token.%
5171     \let\eql@parseopt@next\eql@parseopt@punctpass
5172   \fi
5173   \ifx\eql@parseopt@token,%
5174     \let\eql@parseopt@next\eql@parseopt@punctpass
5175   \fi
5176   \ifx\eql@parseopt@token~%
5177     \let\eql@parseopt@next\eql@parseopt@punctpass
5178   \fi
5179   \ifx\eql@parseopt@token'
5180     \let\eql@parseopt@next\eql@parseopt@punctblock
5181   \fi
5182   \ifx\eql@parseopt@token/%
5183     \let\eql@parseopt@next\eql@parseopt@par
5184   \fi
5185 }

```

`\eql@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:

```

5186 \def\eql@equations@processopt{%
5187   \let\eql@tags@container@block\eql@tags@container@clear
5188   \let\eql@tags@frame@cmd\@firstofone
5189   \let\eql@skip@force@above\@undefined
5190   \let\eql@skip@force@below\@undefined
5191   \let\eql@skip@force@leave\@undefined
5192   \let\eql@display@linewidth\@undefined
5193   \let\eql@display@marginleft\@undefined
5194   \let\eql@display@marginright\@undefined
5195   \eql@abovespace@\z@skip
5196   \eql@belowspace@\z@skip
5197   \eql@displaybreak@prepen@\@MM
5198   \eql@displaybreak@postpen@\@MM
5199   \eql@nextopt@process{equations}%
5200   \let\eql@punct@next\@undefined
5201   \eql@indent@\glueexpr\eql@indent@val\relax
5202   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
5203 }

```

15.2 Single-Line Main

In the following, we define the main routine for the single-line equation mode.

`\eq@single@cr@error` Cannot use line breaks, produce an error message:

```
5204 \def\eq@single@cr@error{%
5205   \eq@error{Cannot use '\string\\' within display equation.
5206     Please switch to equations environment}}%
5207 }
```

`\eq@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:

```
5208 \def\eq@single@start{%
5209   \eq@display@enter
5210   \eq@tagging@start
5211   \eq@dollar@dollar@begin
5212   \eq@display@adjust
5213   \eq@numbering@init
5214   \eq@stack@save@equations
5215   \eq@numbering@single@init
5216   \ifdefined\eq@single@cr@mode
5217     \let\\ \eq@single@cr@mode
5218   \fi
5219   \ifdefined\eq@amp@mode
5220     \let& \eq@break@amp
5221   \fi
5222   \ifdefined\eq@single@native
5223     \let\eq@single@start@sel\eq@single@start@native
5224     \let\eq@single@end@sel\eq@single@end@native
5225   \else
5226     \let\eq@single@start@sel\eq@single@start@print
5227     \let\eq@single@end@sel\eq@single@end@print
5228   \fi
5229   \eq@single@start@sel
5230 }
```

`\eq@single@end` Closing code for single-line equation. Apply punctuation for the block, perform mode-specific ending, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```
5231 \def\eq@single@end{%
5232   \eq@punct@apply@block
5233   \eq@hook@eqout
5234   \eq@single@end@sel
5235   \global\eq@punct@top@reset
5236   \eq@stack@restore
5237   \eq@dollar@dollar@end
5238   \eq@tagging@end
5239   \eq@display@leave
5240 }
```

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

```
5241 \def\eq@single@main{%
```

```

5242 \eql@single@start
5243 \eql@scan@body
5244 \eql@single@end
5245 }

```

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

```

5246 \def\eql@equations@single@set{%
5247   \ifdefined\eql@single@doscan
5248     \let\eql@equations@main\eql@single@main
5249   \else
5250     \let\eql@equations@main\@undefined
5251   \fi
5252 }

```

15.3 Multi-Line Main

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

```

5253 \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

```

5254 \def\eql@multi@main{%
5255   \eql@display@enter
5256   \eql@tagging@start
5257   \eql@dollar@begin
5258   \eql@display@adjust
5259   \eql@numbering@init
5260   \eql@stack@save@equations
5261   \ifdefined\eql@transpose@active
5262     \ifdefined\eql@multi@linesmode\else
5263       \eql@transpose
5264     \fi
5265   \fi
5266   \ifdefined\eql@numbering@subeq@use
5267     \eql@numbering@subeq@init
5268   \fi
5269   \eql@display@init
5270   \let\intertext\eql@intertext
5271   \let\endintertext\endeql@intertext
5272   \eql@shape@align@enable

```

Now measure the given multi-line equations body:

```

5273   \ifdefined\eql@multi@linesmode
5274     \eql@lines@measure
5275   \else
5276     \ifdefined\eql@ampproof@active
5277       \eql@ampproof
5278     \fi
5279     \eql@columns@measure
5280   \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`:

```

5281 \ifdefined\eql@numbering@subeq@use
5282   \eql@numbering@subeq@test
5283 \fi
5284 \ifdefined\eql@multi@linesmode\else
5285   \ifdefined\eql@multi@linesfallback
5286     \ifnum\eql@totalcolumns@=\@ne
5287       \let\eql@multi@linesmode\eql@true
5288       \ifx\eql@multi@linesfallback\z@\else
5289         \eql@lines@measure
5290       \fi
5291     \fi
5292   \fi
5293 \fi

```

Adjust the multi-line equations body:

```

5294 \ifdefined\eql@multi@linesmode
5295   \eql@lines@adjust
5296 \else
5297   \eql@columns@adjust
5298 \fi

```

Now print the multi-line equations body:

```

5299 \eql@display@print
5300 \eql@numbering@print@init
5301 \ifdefined\eql@multi@linesmode
5302   \eql@lines@print
5303 \else
5304   \eql@columns@print
5305 \fi
5306 \eql@display@close

```

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

```

5307 \ifdefined\eql@numbering@subeq@use
5308   \eql@numbering@subeq@close
5309 \fi
5310 \global\eql@punct@top@reset
5311 \eql@stack@restore
5312 \eql@dollar@dollar@end
5313 \eql@tagging@end
5314 \eql@display@leave
5315 }

```

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

```

5316 \def\eql@equations@columns@set{%
5317   \let\eql@equations@main\eql@multi@main
5318   \let\eql@multi@linesmode\eql@false
5319 }
5320 \def\eql@equations@lines@set{%
5321   \let\eql@equations@main\eql@multi@main
5322   \let\eql@multi@linesmode\eql@true
5323 }

```


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`:

```

5324 \newenvironment{equations}{%
5325   \ifmmode
5326     \expandafter\eq@equations@env@cancel
5327   \else
5328     \eq@verbose@info\eq@verbose@msg@enterenv
5329     \expandafter\eq@equations@env@open
5330   \fi
5331 }{%
5332   \ifdefined\eq@equations@main\else
5333     \expandafter\eq@single@end
5334   \fi
5335   \ignorespacesafterend
5336   \eq@verbose@info\eq@verbose@msg@leaveenv
5337 }
5338 \eq@markline@amsthm@register{equations}
5339 \eq@tagging@register@luamml{equations}

```

`\eq@equations@env@cancel`

```

5340 \def\eq@equations@env@cancel{%
5341   \eq@error@mathmode{\string\begin{\@currenvir}}}%
5342   \let\eq@scan@call\eq@scan@env@cancel
5343   \eq@scan@env
5344 }

```

`\eq@equations@env@open`

```

5345 \def\eq@equations@env@open{%
5346   \ifdefined\eq@equations@env@modifier
5347     \eq@equations@test@setall
5348   \else
5349     \eq@equations@nomodifier
5350   \fi
5351   \eq@ampprotect\eq@equations@test\eq@equations@env@start
5352 }

```

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

```

5353 \def\eq@equations@env@start{%
5354   \eq@equations@processopt
5355   \ifdefined\eq@equations@main
5356     \eq@equations@call@set
5357     \expandafter\eq@scan@env
5358   \else
5359     \expandafter\eq@single@start
5360   \fi
5361 }

```

`\eql@equations@call`

```

5362 \def\eql@equations@call{\eql@equations@main\eql@scan@end}
5363 \def\eql@equations@call@test{%
5364   \eql@ampprotect\eql@equations@end@testall\eql@equations@call@apply}
5365 \def\eql@equations@call@apply{\eql@nextopt@apply{equations}\eql@equations@call}
5366 \def\eql@equations@call@set{%
5367   \ifdefined\eql@equations@end@modifier
5368     \let\eql@scan@call\eql@equations@call@test
5369   \else
5370     \let\eql@scan@call\eql@equations@call
5371   \fi
5372 }
```

Square Brackets.

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

```

5373 \newenvironment{equations@sqr}{}{}
5374 \eql@markline@amsthm@register{equations@sqr}
5375 \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`:

```

5376 \def\eql@equations@sqr@open{%
5377   \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
5378   \begin{equations@sqr}%
5379   \eql@verbose@info\eql@verbose@msg@enterenv
5380   \let\]\eql@equations@sqr@close
5381   \ifdefined\eql@equations@sqr@modifier
5382     \eql@equations@test@setall
5383   \else
5384     \eql@equations@nomodifier
5385   \fi
5386   \eql@ampprotect\eql@equations@test\eql@equations@sqr@start
5387 }
```

`\eql@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`:

```

5388 \def\eql@equations@sqr@start{%
5389   \eql@equations@processopt
5390   \ifdefined\eql@equations@main
5391     \eql@equations@call@set
5392     \expandafter\eql@scan@sqr
5393   \else
5394     \expandafter\eql@single@start
5395   \fi
5396 }
```

`\eql@equations@sqr@close` Definition for ‘`]`’. Parse modifiers following ‘`]`’ if not yet processed during scanning and hand on to `\eql@equations@sqr@end`:

```

5397 \protected\def\eql@equations@sqr@close{%
5398   \ifdefined\eql@equations@main
```

```

5399 \let\eqlequations@end@modifier\eql@false
5400 \fi
5401 \ifdefined\eqlequations@end@modifier
5402 \expandafter\eql@ampprotect\expandafter\eqlequations@end@testall
5403 \fi
5404 \eqlequations@sqr@end
5405 }

```

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

```

5406 \def\eqlequations@sqr@end{%
5407 \ifdefined\eqlequations@main\else
5408 \expandafter\eql@single@end
5409 \fi
5410 \eql@verbose@info\eql@verbose@msg@leaveenv
5411 \end{equations@sqr}%
5412 \ignorespaces
5413 }

```

Angle Brackets.

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

```

5414 \newenvironment{equations@ang}{}{}
5415 \eql@markline@amsthm@register{equations@ang}
5416 \eql@tagging@register@luamml{equations@ang}

```

`\eqlequations@ang@open`

```

5417 \def\eqlequations@ang@open{%
5418 \expandafter\eqnaddopt\expandafter{\eqlequations@ang@opt}%
5419 \begin{equations@ang}%
5420 \eql@verbose@info\eql@verbose@msg@enterenv
5421 \let\>\eqlequations@ang@close
5422 \ifdefined\eqlequations@ang@modifier
5423 \eqlequations@test@setall
5424 \else
5425 \eqlequations@nomodifier
5426 \fi
5427 \eql@ampprotect\eqlequations@test\eqlequations@ang@start
5428 }

```

`\eqlequations@ang@start` Process arguments and start handling the equation:

```

5429 \def\eqlequations@ang@start{%
5430 \eqlequations@processopt
5431 \ifdefined\eqlequations@main
5432 \eqlequations@call@set
5433 \expandafter\eql@scan@ang
5434 \else
5435 \expandafter\eql@single@start
5436 \fi
5437 }

```

`\eqlequations@ang@close` **TODO:** describe

```

5438 \def\eqlequations@ang@close{%
5439 \ifdefined\eqlequations@main

```

```

5440 \let\eql@equations@end@modifier\eql@false
5441 \fi
5442 \ifdefined\eql@equations@end@modifier
5443 \expandafter\eql@ampprotect\expandafter\eql@equations@end@testall
5444 \fi
5445 \eql@equations@ang@end
5446 }

```

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

```

5447 \def\eql@equations@ang@end{%
5448 \ifdefined\eql@equations@main\else
5449 \expandafter\eql@single@end
5450 \fi
5451 \eql@verbose@info\eql@verbose@msg@leaveenv
5452 \end{equations@ang}%
5453 \ignorespaces
5454 }

```

16 Options

16.1 Selection Tools

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

```

5455 \def\eql@decide@abovebelow#1#2#3#4#5{%
5456 \eql@decide@select{#1}{#2}{#3}{%
5457 {abovebelow,both,tb}{#4#5},%
5458 {above,top,t}{#4},%
5459 {below,bottom,b}{#5}}

```

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

```

5460 \def\eql@decide@situation#1#2#3#4{%
5461 \eql@decide@select{#1}{#2}{#3}{%
5462 {{long}}{\def#4{0}}},%
5463 {{short}}{\def#4{1}}},%
5464 {{cont}}{\def#4{2}}},%
5465 {{par}}{\def#4{3}}},%
5466 {{top}}{\def#4{4}}},%
5467 {{noskip}}{\def#4{5}}},%
5468 {{medskip}}{\def#4{6}}}}

```

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

```

5469 \def\eql@decide@delim#1#2#3{%
5470 \eql@decide@select{#1}{#2}{#3}{%
5471 {{,.,\eql@decide@false}}{\eql@box@wrap}}},%
5472 {{\eql@decide@true,r,p,round,parentheses}}{\eql@box@delim()}},%
5473 {{s,b,sqr,square}}{\eql@box@delim[]}},%
5474 {{c,B,curly,braces}}{\eql@box@delim\lbrace\rbrace}},%
5475 {{a,ang,angle}}{\eql@box@delim\langle\rangle}},%
5476 {{v,vert}}{\eql@box@delim\vert\vert}},%
5477 {{d,V,dvert}}{\eql@box@delim\Vert\Vert}},%
5478 {\relax\eql@box@delim#3}}}%

```

TODO: describe

```
5479 \def\eql@keyall{equations,box,setup}
```

16.2 Options Declarations

We now declare all key-value pairs for options sorted by their category.

Modes for Equations Box Environment. Declare horizontal and vertical alignment modes for the boxed equations environment. Also declare spacing of columns:

```
5480 \eql@define@key{box}{lines,ln,gathered,gather,ga}[]{%
5481   \eql@box@lines@set}
5482 \eql@define@key{box}{columns,col,aligned,align,al}[]{%
5483   \eql@box@columns@set}
5484 \eql@define@key{box}{cases}[]{%
5485   \eql@box@cases@set\eql@box@ldelim\lbrace%
5486   \def\eql@box@colsep{\eql@box@condsep}}
5487 \eql@define@key{box}{matrix,pmatrix}[r]{%
5488   \eql@box@lines@set\eql@shape@set{center}%
5489   \let\eql@spread@reset\eql@true\def\eql@spread@val{\z@}%
5490   \def\eql@box@colsep{\eql@box@matrixsep}%
5491   \let\eql@mathstyle\@empty
5492   \eql@punct@clear
5493   \eql@box@cr@test@setopt
5494   \eql@box@amp@test@setopt
5495   \let\eql@box@end@modifier\eql@false
5496   \eql@decide@delim{#3}{#2}{#1}}
5497 \eql@define@key{box}{top,t}[]{\let\eql@box@box\vtop}
5498 \eql@define@key{box}{center,c}[]{\let\eql@box@box\vcenter}
5499 \eql@define@key{box}{bottom,b}[]{\let\eql@box@box\vbox}
5500 \eql@define@key{box}{intro}{%
5501   \def\eql@box@cases@condintro{#1}}
5502 \eql@define@key{box}{introtext}{%
5503   \def\eql@box@cases@condintro{%
5504     \ifmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}
5505 \eql@define@key{box}{textcond}[true]{%
5506   \eql@decide@select{#3}{#2}{#1}{%
5507     {\eql@decide@true,text}{\let\eql@box@cases@condtext\eql@true}},%
5508     {\eql@decide@false,math}{\let\eql@box@cases@condtext\eql@false}}}}
5509 \eql@define@key{setup}{scanbox}[true]{%
5510   \eql@decide@bool{#3}{#2}{#1}\eql@box@doscan}
5511 \eql@define@key{box}{scan}[true]{%
5512   \eql@decide@bool{#3}{#2}{#1}\eql@box@doscan}
5513 \eql@define@key{setup}{boxangopt}[]{%
5514   \def\eql@box@ang@opt{columns,#1}}
```

Modes for Equations Environment.

```
5515 \let\eql@box@doscan\eqn@false
```

Declare modes and switches for the equations environment:

```
5516 \eql@define@key{equations}{single,1,equation,eq}[]{\eql@equations@single@set}
5517 \eql@define@key{equations}{lines,ln,gathered,gather,ga}[]{%
5518   \eql@equations@lines@set}
5519 \eql@define@key{equations}{columns,col,aligned,align,al}[]{%
5520   \eql@equations@columns@set}
```

```

5521 \eql@define@key{equations,setup}{transpose}[true]{%
5522   \eql@decide@select{#3}{#2}{#1}{%
5523     {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
5524     {\noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
5525     {\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}%
5526 \eql@define@key{setup}{nativeequation}[true]{%
5527   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
5528 \eql@define@key{equations}{native}[true]{%
5529   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
5530   \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
5531 \eql@define@key{setup}{scanequation}[true]{%
5532   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscanscan}
5533 \eql@define@key{equations}{scan}[true]{%
5534   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscanscan}
5535   \ifx\eql@equations@main\eql@multi@main\else\eql@equations@single@set\fi}
5536 \eql@define@key{setup}{sqropt}[]{%
5537   \def\eql@equations@sqr@opt{equation,#1}}
5538 \eql@define@key{setup}{angopt}[]{%
5539   \def\eql@equations@ang@opt{columns,#1}}

```

TODO: describe

```

5540 \eql@define@key{control}{restoreexterior}[]{\eql@display@restore}
5541 \eql@define@key{control}{restoreexterior*}[]{\@arrayparboxrestore}

```

Modes for Modifier Scanning.

```

5542 \def\eql@equations@nomodifier{%
5543   \eql@equations@test@setopt
5544   \let\eql@equations@end@modifier\eql@false
5545   \eql@multi@cr@test@setopt
5546   \eql@columns@amp@test@setopt
5547   \eql@break@cr@test@setopt
5548   \eql@break@amp@test@setopt
5549 }
5550 \let\eql@equations@env@modifier\eql@false
5551 \let\eql@equations@sqr@modifier\eql@true
5552 \let\eql@equations@ang@modifier\eql@true
5553 \let\eql@equations@end@modifier\eql@true
5554 \eql@equations@test@setall
5555 \eql@multi@cr@test@setall
5556 \eql@columns@amp@test@setall
5557 \eql@break@cr@test@setall
5558 \eql@break@amp@test@setall
5559 \eql@break@test@setopt
5560 \let\eql@multi@cr@relnext\eql@true

5561 \def\eql@box@nomodifier{%
5562   \eql@box@test@setopt
5563   \let\eql@box@end@modifier\eql@false
5564   \eql@box@cr@test@setopt
5565   \eql@box@amp@test@setopt
5566 }
5567 \let\eql@box@env@modifier\eql@false
5568 \let\eql@box@ang@modifier\eql@true
5569 \let\eql@box@end@modifier\eql@true
5570 \eql@box@test@setall
5571 \eql@box@cr@test@setall
5572 \eql@box@amp@test@setall

```

```

5573 \eqld@define@key{setup}{modifier}[true]{%
5574   \eqld@decide@if{#3}{#2}{#1}%
5575   {\let\eqld@equations@env@modifier\eqld@true
5576    \let\eqld@equations@sqr@modifier\eqld@true
5577    \let\eqld@equations@ang@modifier\eqld@true
5578    \let\eqld@box@env@modifier\eqld@true
5579    \let\eqld@box@ang@modifier\eqld@true
5580    \eqld@multi@cr@test@setall\eqld@break@cr@test@setall
5581    \eqld@columns@amp@test@setall\eqld@break@amp@test@setall
5582    \eqld@box@cr@test@setall\eqld@box@amp@test@setall
5583    \eqld@break@test@setall}%
5584   {\let\eqld@equations@env@modifier\eqld@false
5585    \let\eqld@equations@sqr@modifier\eqld@false
5586    \let\eqld@equations@ang@modifier\eqld@false
5587    \let\eqld@box@env@modifier\eqld@false
5588    \let\eqld@box@ang@modifier\eqld@false
5589    \eqld@multi@cr@test@setopt\eqld@break@cr@test@setopt
5590    \eqld@columns@amp@test@setopt\eqld@break@amp@test@setopt
5591    \eqld@box@cr@test@setopt\eqld@box@amp@test@setopt
5592    \eqld@break@test@setopt}}
5593 \eqld@define@key{setup}{modifierenv}[true]{%
5594   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@env@modifier
5595   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@env@modifier}
5596 \eqld@define@key{setup}{modifiersqrang}[true]{%
5597   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@sqr@modifier
5598   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@ang@modifier
5599   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@ang@modifier}
5600 \eqld@define@key{setup}{modifierend}[true]{%
5601   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@end@modifier
5602   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@end@modifier}
5603 \eqld@define@key{setup}{modifierreqncr}[true]{\eqld@decide@if{#3}{#2}{#1}%
5604   {\eqld@multi@cr@test@setall\eqld@break@cr@test@setall
5605    \eqld@box@cr@test@setall}%
5606   {\eqld@multi@cr@test@setopt\eqld@break@cr@test@setopt
5607    \eqld@box@cr@test@setopt}}
5608 \eqld@define@key{setup}{modifierreqnamp}[true]{\eqld@decide@if{#3}{#2}{#1}%
5609   {\eqld@columns@amp@test@setall\eqld@break@amp@test@setall
5610    \eqld@box@amp@test@setall}%
5611   {\eqld@columns@amp@test@setopt\eqld@break@amp@test@setopt
5612    \eqld@box@amp@test@setopt}}
5613 \eqld@define@key{setup}{modifierbreak}[true]{\eqld@decide@if{#3}{#2}{#1}%
5614   \eqld@break@test@setall\eqld@break@test@setopt}
5615 \eqld@define@key{setup}{modifierwarning}[all]{%
5616   \eqld@decide@select{#3}{#2}{#1}{%
5617     {\eqld@decide@false{\let\eqld@parseopt@warn@env\empty
5618      \let\eqld@parseopt@warn@cr\empty}},%
5619     {\env,-}{\let\eqld@parseopt@warn@env\eqld@warn@parseopt
5620      \let\eqld@parseopt@warn@cr\empty}},%
5621     {\all,\eqld@decide@true}{\let\eqld@parseopt@warn@env\eqld@warn@parseopt
5622      \let\eqld@parseopt@warn@cr\eqld@warn@parseopt}},%
5623     {\verbose,+}{\let\eqld@parseopt@warn@env\eqld@warn@parseopt@verbose
5624      \let\eqld@parseopt@warn@cr\eqld@warn@parseopt@verbose}}}}
5625 \eqld@define@key{setup}{modifierreqn}[true]{%
5626   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@env@modifier
5627   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@sqr@modifier
5628   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@ang@modifier}
5629 \eqld@define@key{setup}{modifierreqnenv}[true]{%
5630   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@env@modifier}

```

```

5631 \eqld@define@key{setup}{modifiereqnsqr}[true]{%
5632   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@sqr@modifier}
5633 \eqld@define@key{setup}{modifiereqnang}[true]{%
5634   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@ang@modifier}
5635 \eqld@define@key{setup}{modifiereqnend}[true]{%
5636   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@end@modifier}
5637 \eqld@define@key{setup}{modifiereqncr}[true]{\eqld@decide@if{#3}{#2}{#1}%
5638   {\eqld@multi@cr@test@setall\eqld@break@cr@test@setall}%
5639   {\eqld@multi@cr@test@setopt\eqld@break@cr@test@setopt}}
5640 \eqld@define@key{setup}{modifiereqnamp}[true]{\eqld@decide@if{#3}{#2}{#1}%
5641   {\eqld@columns@amp@test@setall\eqld@break@amp@test@setall}%
5642   {\eqld@columns@amp@test@setopt\eqld@break@amp@test@setopt}}

5643 \eqld@define@key{setup}{modifierbox}[true]{%
5644   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@env@modifier
5645   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@ang@modifier}
5646 \eqld@define@key{setup}{modifierboxenv}[true]{%
5647   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@env@modifier}
5648 \eqld@define@key{setup}{modifierboxang}[true]{%
5649   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@ang@modifier}
5650 \eqld@define@key{setup}{modifierboxend}[true]{%
5651   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@end@modifier}
5652 \eqld@define@key{setup}{modifierboxcr}[true]{%
5653   \eqld@decide@if{#3}{#2}{#1}\eqld@box@cr@test@setall\eqld@box@cr@test@setopt}
5654 \eqld@define@key{setup}{modifierboxamp}[true]{%
5655   \eqld@decide@if{#3}{#2}{#1}\eqld@box@amp@test@setall\eqld@box@amp@test@setopt}

5656 \eqld@define@key{equations}{modifierend}[true]{%
5657   \eqld@decide@bool{#3}{#2}{#1}\eqld@equations@end@modifier}
5658 \eqld@define@key{equations}{modifiercr}[true]{\eqld@decide@if{#3}{#2}{#1}%
5659   {\eqld@multi@cr@test@setall\eqld@break@cr@test@setall}%
5660   {\eqld@multi@cr@test@setopt\eqld@break@cr@test@setopt}}
5661 \eqld@define@key{equations}{modifieramp}[true]{\eqld@decide@if{#3}{#2}{#1}%
5662   {\eqld@columns@amp@test@setall\eqld@break@amp@test@setall}%
5663   {\eqld@columns@amp@test@setopt\eqld@break@amp@test@setopt}}
5664 \eqld@define@key{setup,equations}{crrelnext}[true]{%
5665   \eqld@decide@bool{#3}{#2}{#1}\eqld@multi@cr@relnext}

5666 \eqld@define@key{box}{modifier}[true]{\eqld@decide@if{#3}{#2}{#1}%
5667   {\eqld@box@cr@test@setall\eqld@box@amp@test@setall
5668     \let\eqld@box@end@modifier\eqld@true}%
5669   {\eqld@box@cr@test@setopt\eqld@box@amp@test@setopt
5670     \let\eqld@box@end@modifier\eqld@false}}
5671 \eqld@define@key{box}{modifierend}[true]{%
5672   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@end@modifier}
5673 \eqld@define@key{box}{modifiercr}[true]{%
5674   \eqld@decide@if{#3}{#2}{#1}\eqld@box@cr@test@setall\eqld@box@cr@test@setopt}
5675 \eqld@define@key{box}{modifieramp}[true]{%
5676   \eqld@decide@if{#3}{#2}{#1}\eqld@box@amp@test@setall\eqld@box@amp@test@setopt}

5677 \eqld@define@key\eqld@keyall{spread}{%
5678   \let\eqld@spread@reset\eqld@false\def\eqld@spread@val{#1}}
5679 \eqld@define@key\eqld@keyall{spread*}[Opt]{%
5680   \let\eqld@spread@reset\eqld@true\def\eqld@spread@val{#1}}
5681 \eqld@define@key\eqld@keyall{strut}[true]{\eqld@decide@select{#3}{#2}{#1}{%
5682   {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%

```

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


```

5683    {{cell}}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
5684    {{tag}}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
5685    {\eql@decide@true
5686     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
5687 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

5688 \eql@define@key\eql@keyall{displayheight}[strut]{%
5689   \eql@decide@select{#3}{#2}{#1}{%
5690     {\eql@decide@false{\let\eql@display@height\@undefined}},%
5691     {{strut}}{\def\eql@display@height{\ht\eql@strutbox@}}},%
5692     {\relax}{\def\eql@display@height{#1}}}}
5693 \eql@define@key\eql@keyall{displaydepth}[strut]{%
5694   \eql@decide@select{#3}{#2}{#1}{%
5695     {\eql@decide@false{\let\eql@display@depth\@undefined}},%
5696     {{strut}}{\def\eql@display@depth{\dp\eql@strutbox@}}},%
5697     {\relax}{\def\eql@display@depth{#1}}}}

```

Settings concerning page breaks:

```

5698 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5699   {{force,4,\eql@decide@true}}{\eql@displaybreak@pre4}},%
5700   {{high,3}}{\eql@displaybreak@pre3}},%
5701   {{med,medium,2}}{\eql@displaybreak@pre2}},%
5702   {{low,1}}{\eql@displaybreak@pre1}},%
5703   {{0,\eql@decide@false}}{\eql@displaybreak@pre0}},%
5704   {{default,inherit,-1}}{\eql@displaybreak@pre\m@ne}}}}
5705 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5706   {{force,4,\eql@decide@true}}{\eql@displaybreak@post4}},%
5707   {{high,3}}{\eql@displaybreak@post3}},%
5708   {{med,medium,2}}{\eql@displaybreak@post2}},%
5709   {{low,1}}{\eql@displaybreak@post1}},%
5710   {{0,\eql@decide@false}}{\eql@displaybreak@post0}},%
5711   {{default,inherit,-1}}{\eql@displaybreak@post\m@ne}}}}
5712 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
5713   \eql@decide@select{#3}{#2}{#1}{%
5714     {{full,4}}{\eql@displaybreak@inter4}},%
5715     {{high,3}}{\eql@displaybreak@inter3}},%
5716     {{med,medium,2}}{\eql@displaybreak@inter2}},%
5717     {{low,1}}{\eql@displaybreak@inter1}},%
5718     {{0,\eql@decide@false}}{\eql@displaybreak@inter\z@}}}}
5719 \eql@define@key{equations}{prepenalty}{%
5720   \eql@displaybreak@prepen\numexpr#1\relax}
5721 \eql@define@key{equations}{postpenalty}{%
5722   \eql@displaybreak@postpen\numexpr#1\relax}
5723 \eql@define@key{equations,setup}{interpenalty}{%
5724   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

5725 \eql@define@key{control}{vspace}[]{\eql@vspace@add{#1}}
5726 \eql@define@key{control}{vspace*}[]{\eql@vspace@addfixedbefore{#1}}
5727 \eql@define@key{control}{vspace!}[]{\eql@vspace@addfixedafter{#1}}
5728 \eql@define@key{control}{break}[4]{\eql@displaybreak@level[#{#1}]}
5729 \eql@define@key{control}{penalty}[]{\eql@displaybreak@star{#1}}

```

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

```

5730 \eql@define@key{equations}{noskip}[both]{%

```

```

5731 \eql@decide@abovebelow{#3}{#2}{#1}%
5732   {\def\eql@skip@force@above{5}}%
5733   {\def\eql@skip@force@below{5}}%
5734 \eql@define@key{equations}{short}[above]{%
5735   \eql@decide@abovebelow{#3}{#2}{#1}%
5736   {\def\eql@skip@force@above{1}}%
5737   {\def\eql@skip@force@below{1}}%
5738 \eql@define@key{equations}{long}[both]{%
5739   \eql@decide@abovebelow{#3}{#2}{#1}%
5740   {\def\eql@skip@force@above{0}}%
5741   {\def\eql@skip@force@below{0}}%
5742 \eql@define@key{equations}{medskip}[both]{%
5743   \eql@decide@abovebelow{#3}{#2}{#1}%
5744   {\def\eql@skip@force@above{6}}%
5745   {\def\eql@skip@force@below{6}}%
5746 \eql@define@key{equations}{par}[par]{%
5747   \eql@decide@select{#3}{#2}{#1}{%
5748     {\default,\eql@decide@false}{\let\eql@skip@force@leave\@undefined}},%
5749     {\cont,hmode}{\let\eql@skip@force@leave\z@}},%
5750     {\par,vmode}{\let\eql@skip@force@leave\@ne
5751       \ifdefined\eql@skip@force@below\else
5752         \def\eql@skip@force@below{3}%
5753       \fi}},%
5754     {\top}{\let\eql@skip@force@leave\tw@
5755       \ifdefined\eql@skip@force@below\else
5756         \def\eql@skip@force@below{4}%
5757       \fi}}}}

```

Specify vertical spacing explicitly:

```

5758 \eql@define@key{equations}{skip}{%
5759   \def\eql@skip@force@above{7}%
5760   \def\eql@skip@custom@above{#1}%
5761   \let\eql@skip@force@below\eql@skip@force@above
5762   \let\eql@skip@custom@below\eql@skip@custom@above}
5763 \eql@define@key{equations}{aboveskip}{%
5764   \def\eql@skip@force@above{7}%
5765   \def\eql@skip@custom@above{#1}}
5766 \eql@define@key{equations}{belowskip}{%
5767   \def\eql@skip@force@below{7}%
5768   \def\eql@skip@custom@below{#1}}
5769 \eql@define@key{equations}{abovespace}{%
5770   \advance\eql@abovespace@glueexpr#1\relax}
5771 \eql@define@key{equations}{belowspace}{%
5772   \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

5773 \eql@define@key{intertext}{skip}{%
5774   \def\eql@skip@force@above{7}%
5775   \def\eql@skip@custom@above{#1}%
5776   \let\eql@skip@force@below\eql@skip@force@above
5777   \let\eql@skip@custom@below\eql@skip@custom@above}
5778 \eql@define@key{intertext}{aboveskip}{%
5779   \def\eql@skip@force@below{7}%
5780   \def\eql@skip@custom@below{#1}}
5781 \eql@define@key{intertext}{belowskip}{%
5782   \def\eql@skip@force@above{7}%
5783   \def\eql@skip@custom@above{#1}}
5784 \eql@define@key{intertext}{noskip}[both]{%

```

```

5785 \eql@decide@abovebelow{#3}{#2}{#1}%
5786   {\def\eql@skip@force@below{5}}%
5787   {\def\eql@skip@force@above{5}}%
5788 \eql@define@key{intertext}{short}[both]{%
5789   \eql@decide@abovebelow{#3}{#2}{#1}%
5790   {\def\eql@skip@force@below{1}}%
5791   {\def\eql@skip@force@above{1}}%
5792 \eql@define@key{intertext}{long}[both]{%
5793   \eql@decide@abovebelow{#3}{#2}{#1}%
5794   {\def\eql@skip@force@below{0}}%
5795   {\def\eql@skip@force@above{0}}%
5796 \eql@define@key{intertext}{medskip}[both]{%
5797   \eql@decide@abovebelow{#3}{#2}{#1}%
5798   {\def\eql@skip@force@below{6}}%
5799   {\def\eql@skip@force@above{6}}%

```

Configure general vertical spacing behaviour for various situations:

```

5800 \eql@define@key{setup}{skip,longskip}{%
5801   \abovedisplayskip\glueexpr#1\relax
5802   \belowdisplayskip\abovedisplayskip
5803   \def\eql@skip@long@above{#1}%
5804   \let\eql@skip@long@below\eql@skip@long@above}
5805 \eql@define@key{setup}{aboveskip,abovelongskip}{%
5806   \abovedisplayskip\glueexpr#1\relax
5807   \def\eql@skip@long@above{#1}}
5808 \eql@define@key{setup}{belowskip,belowlongskip}{%
5809   \belowdisplayskip\glueexpr#1\relax
5810   \def\eql@skip@long@below{#1}}
5811 \eql@define@key{setup}{aboveshortskip}{%
5812   \abovedisplayshortskip\glueexpr#1\relax
5813   \def\eql@skip@short@above{#1}}
5814 \eql@define@key{setup}{belowshortskip}{%
5815   \belowdisplayshortskip\glueexpr#1\relax
5816   \def\eql@skip@short@below{#1}}
5817 \eql@define@key{setup}{tagskip}{%
5818   \def\eql@skip@tag@above{#1}%
5819   \let\eql@skip@tag@below\eql@skip@tag@above}
5820 \eql@define@key{setup}{abovetagskip}{%
5821   \def\eql@skip@tag@above{#1}}
5822 \eql@define@key{setup}{belowtagskip}{%
5823   \def\eql@skip@tag@below{#1}}
5824 \eql@define@key{setup}{medskip}{%
5825   \def\eql@skip@med@above{#1}%
5826   \let\eql@skip@med@below\eql@skip@med@above}
5827 \eql@define@key{setup}{abovemedskip}{%
5828   \def\eql@skip@med@above{#1}}
5829 \eql@define@key{setup}{belowmedskip}{%
5830   \def\eql@skip@med@below{#1}}
5831 \eql@define@key{setup}{abovetopskip}{%
5832   \def\eql@skip@top@above{#1}}
5833 \eql@define@key{setup}{belowtopskip}{%
5834   \def\eql@skip@top@below{#1}}
5835 \eql@define@key{setup}{aboveparskip}{%
5836   \def\eql@skip@par@above{#1}}
5837 \eql@define@key{setup}{belowparskip}{%
5838   \def\eql@skip@par@below{#1}}
5839 \eql@define@key{setup}{abovecontskip}{%
5840   \eql@decide@select{#3}{#2}{#1}{%

```

```

5841    {{hide}}{\def\eq@skip@cont@above{\eq@spread@val-\eq@skip@long@below}}},%
5842    {\relax{\def\eq@skip@cont@above{#1}}}}
5843 \eq@define@key{setup}{belowcontskip}{%
5844   \def\eq@skip@cont@below{#1}}
5845 \eq@define@key{setup}{shortmode}{%
5846   \eq@decide@select{#3}{#2}{#1}{%
5847     {\eq@decide@false,never}{\def\eq@skip@mode@short{0}}},%
5848     {{above,neverbelow,belowoff}{\def\eq@skip@mode@short{1}}},%
5849     {{belowone,belowsingle}{\def\eq@skip@mode@short{2}}},%
5850     {{belowall,always,on}{\def\eq@skip@mode@short{3}}}}
5851 \eq@define@key{setup}{abovecontmode}{%
5852   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
5853 \eq@define@key{setup}{belowcontmode}{%
5854   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
5855 \eq@define@key{setup}{aboveparmode}{%
5856   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@above}
5857 \eq@define@key{setup}{belowparmode}{%
5858   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@below}
5859 \eq@define@key{setup}{abovetopmode}{%
5860   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@above}
5861 \eq@define@key{setup}{belowtopmode}{%
5862   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@below}

```

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

```

5863 \eq@define@key{equations,subequations}{label}{\eq@tags@addblock@label{#1}}
5864 \eq@define@key{equations,subequations}{labelname}{\eq@tags@addblock@name{#1}}
5865 \eq@define@key{equations,subequations}{tag}{\eq@tags@addblock@tag{#1}}
5866 \eq@define@key{equations,subequations}{tag*}{%
5867   \eq@tags@addblock@tagform@off\eq@tags@addblock@tag{#1}}
5868 \eq@define@key{equations,subequations}{taglabel}{\eq@tags@addblock@ref{#1}}

```

TODO: describe

```

5869 \eq@define@key{control}{label}{\eq@tags@add@label{#1}}
5870 \eq@define@key{control}{labelname}{\eq@tags@add@name{#1}}
5871 \eq@define@key{control}{tag}{\eq@tags@add@tag{#1}}
5872 \eq@define@key{control}{tag*}{\eq@tags@add@tagform@off\eq@tags@add@tag{#1}}
5873 \eq@define@key{control}{taglabel}{\eq@tags@add@ref{#1}}
5874 \eq@define@key{control}{shifftag}{\eq@tags@add@raiseshift{#1}}
5875 \eq@define@key{control}{smashtag}{\eq@tags@add@raisesmash{#1}}
5876 \eq@define@key{control}{pushtag}{\eq@tags@add@forceraise}

```

TODO: describe

```

5877 \eq@define@key{setup}{labelname}{\protected@edef\eq@tags@name@generic{#1}}
5878 \eq@define@key{setup}{autolabel}[true]{%
5879   \eq@decide@bool{#3}{#2}{#1}\eq@tags@autolabel}
5880 \eq@define@key{setup}{autotag}[true]{%
5881   \eq@decide@bool{#3}{#2}{#1}\eq@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

5882 \eq@define@key{equations,setup}{tagmargin}[auto]{%
5883   \eq@decide@select{#3}{#2}{#1}{%
5884     {{auto,\eq@decide@false}{\let\eq@tagmargin@val\undefined}}},%
5885     {\relax{\def\eq@tagmargin@val{#1}}}}
5886 \eq@define@key{equations,setup}{tagmargin*}{%
5887   \settowidth\dimen@{#1}\edef\eq@tagmargin@val{\the\dimen@}}

```

```

5888 \eqld@define@key{equations,setup}{tagmarginratio}{%
5889   \eqld@tagmargin@ratio@{\dimexpr#1pt\relax}
5890 \eqld@define@key{equations,setup}{tagmarginthreshold}{%
5891   \def\eqld@tagmargin@threshold{#1}}
5892 \eqld@define@key{equations,setup}{mintagsep}{\def\eqld@tagsepmin@val{#1}}
5893 \eqld@define@key{equations,setup}{mintagwidth}{%
5894   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
5895 \eqld@define@key{equations,setup}{mintagwidth*}{%
5896   \settowidth\eqld@tagwidthmin@{#1}}
5897 \eqld@define@key{equations,setup}{tagsnap}{%
5898   \eqld@decide@select{#3}{#2}{#1}{%
5899     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
5900     {\relax{\def\eqld@tagpos@snap{#1}}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

5901 \eqld@define@key{equations,setup}{tagbox,taglayout}{%
5902   \eqld@tags@taglayout@set{#1}}
5903 \eqld@define@key{equations,setup}{tagbox*,taglayout*}{%
5904   \eqld@tags@taglayout@set@direct{#1}}
5905 \eqld@define@key{equations,setup}{tagform}{%
5906   \eqld@tags@tagform@set{#1}}
5907 \eqld@define@key{equations,setup}{tagform*}{%
5908   \eqld@tags@tagform@set@direct{#1}}
5909 \eqld@define@key{equations,setup}{subeqtemplate}{%
5910   \eqld@subequations@template@set{#1}}

5911 \eqld@define@key{control}{tagbox,taglayout}{%
5912   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
5913 \eqld@define@key{control}{tagbox*,taglayout*}{%
5914   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
5915 \eqld@define@key{control}{tagform}{%
5916   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
5917 \eqld@define@key{control}{tagform*}{\####1}{%
5918   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

5919 \eqld@define@key{equations,setup}{numberline,number,num,numline,n}[all]{%
5920   \eqld@decide@select{#3}{#2}{#1}{%
5921     {\eqld@decide@false,0,*}{\let\eqld@numbering@active\eqld@false}},%
5922     {\eqld@decide@true,!}{\let\eqld@numbering@active\eqld@true}},%
5923     {\none,n,-}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
5924       \let\eqld@numbering@active\eqld@false}},%
5925     {\single,1}{\let\eqld@numbering@mode\eqld@numbering@mode@single
5926       \let\eqld@numbering@active\eqld@true}},%
5927     {\multi,@}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
5928       \let\eqld@numbering@active\eqld@true}},%
5929     {\relax{\eqld@numbering@set{#1}}}}}
5930 \eqld@define@key{equations,setup}{nonumber,nn,*}[]{%
5931   \let\eqld@numbering@active\eqld@false}
5932 \eqld@define@key{equations,setup}{donumber,dn,!}[]{%
5933   \let\eqld@numbering@active\eqld@true}
5934 \eqld@define@key{equations,setup}{tagsleft,leqno}[]{%
5935   \let\eqld@tagsleft\eqld@true}
5936 \eqld@define@key{equations,setup}{tagsright,reqno}[]{%
5937   \let\eqld@tagsleft\eqld@false}
5938 \eqld@define@key{equations,setup}{tags,eqno}{%

```

```

5939 \eqld@decide@select{#3}{#2}{#1}{%
5940   {{right,r}}{\let\eqld@tagsleft\eqld@false}},%
5941   {{left,l}}{\let\eqld@tagsleft\eqld@true}}}}
5942 \eqld@define@key{equations,setup}{evadetag,avoidtag}[true]{%
5943   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}
5944 \eqld@define@key{equations,setup}{tagbetween}[true]{%
5945   \eqld@decide@bool{#3}{#2}{#1}\eqld@tagpos@doconvert}

```

TODO: describe

```

5946 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
5947 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
5948 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
5949 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

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

```

5950 \eqld@define@key{equations,setup}{layout}{\eqld@decide@select{#3}{#2}{#1}{%
5951   {{center,c}}{\let\eqld@layoutleft\eqld@false}},%
5952   {{left,l}}{\let\eqld@layoutleft\eqld@true}}}}
5953 \eqld@define@key{equations,setup}{center}[]{%
5954   \let\eqld@layoutleft\eqld@false}
5955 \eqld@define@key{equations,setup}{flushleft,left}[]{%
5956   \let\eqld@layoutleft\eqld@true}
5957 \eqld@define@key{equations,setup}{leftmargin}{\def\eqld@layoutleftmargin{#1}}
5958 \eqld@define@key{equations,setup}{leftmargin*}{%
5959   \settowidth\dimen@{#1}\edef\eqld@layoutleftmargin{\the\dimen@}}
5960 \eqld@define@key{equations,setup}{minleftmargin}{%
5961   \def\eqld@layoutleftmarginmin{#1}}
5962 \eqld@define@key{equations,setup}{maxleftmargin}{%
5963   \eqld@decide@select{#3}{#2}{#1}{%
5964     {\eqld@decide@false{\def\eqld@layoutleftmarginmax{.5\maxdimen}}},%
5965     {\relax{\def\eqld@layoutleftmarginmax{#1}}}}}
5966 \eqld@define@key{equations,box}{margin}{%
5967   \def\eqld@display@marginleft{#1}\def\eqld@display@marginright{#1}}
5968 \eqld@define@key{equations,box}{marginleft}{\def\eqld@display@marginleft{#1}}
5969 \eqld@define@key{equations,box}{marginright}{\def\eqld@display@marginright{#1}}
5970 \eqld@define@key{equations}{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

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

```

5971 \eqld@define@key{equations,setup}{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5972   {{max,full,4}}{\eqld@alignbadness@inf@bad}},%
5973   {{high,3}}{\eqld@alignbadness@54\relax}},%
5974   {{med,medium,2}}{\eqld@alignbadness@18\relax}},%
5975   {{low,1}}{\eqld@alignbadness@6\relax}},%
5976   {{0,\eqld@decide@false}}{\eqld@alignbadness@z@}}}}
5977 \eqld@define@key{equations,setup}{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5978   {{max,full,4}}{\eqld@tagbadness@inf@bad}},%
5979   {{high,3}}{\eqld@tagbadness@54\relax}},%
5980   {{med,medium,2}}{\eqld@tagbadness@18\relax}},%
5981   {{low,1}}{\eqld@tagbadness@6\relax}},%
5982   {{0,\eqld@decide@false}}{\eqld@tagbadness@z@}}}}
5983 \eqld@define@key{equations,setup}{alignbadness}{%
5984   \eqld@alignbadness@numexpr#1\relax}

```

```

5985 \eqld@define@key{equations,setup}{tagbadness}{%
5986   \eqld@tagbadness@numexpr#1\relax}
5987 \eqld@define@key{equations,setup}{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5988   {0,\eqld@decide@false}{\def\eqld@colsepmin@val{0pt}}},%
5989   {\relax{\def\eqld@colsepmin@val{#1}}}}
5990 \eqld@define@key{equations,setup}{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5991   {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
5992   {0{\def\eqld@colsepmax@val{0pt}}},%
5993   {\relax{\def\eqld@colsepmax@val{#1}}}}
5994 \eqld@define@key{equations,setup}{fulllength}[true]{%
5995   \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

```

TODO: is boxcolsep vs breakcolsep okay??!

```

5996 \eqld@define@key{equations,setup}{linesep}{\eqld@decide@select{#3}{#2}{#1}{%
5997   {0,\eqld@decide@false}{\def\eqld@break@line@sep{0pt}}},%
5998   {\relax{\def\eqld@break@line@sep{#1}}}}
5999 \eqld@define@key{equations,setup}{linesep*}{\eqld@decide@select{#3}{#2}{#1}{%
6000   {0,\eqld@decide@false}{\def\eqld@break@line@shortsep{0pt}}},%
6001   {\relax{\def\eqld@break@line@shortsep{#1}}}}
6002 \eqld@define@key{box,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
6003   {0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
6004   {{short,*}{\def\eqld@box@colsep{\eqld@box@shortsep}}},%
6005   {\relax{\def\eqld@box@colsep{#1}}}}
6006   \let\eqld@break@col@sep\eqld@box@colsep}
6007 \eqld@define@key{equations}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
6008   {0,\eqld@decide@false}{\def\eqld@break@col@sep{0pt}}},%
6009   {\relax{\def\eqld@break@col@sep{#1}}}}
6010   \let\eqld@colsepmin@val\eqld@box@colsep
6011   \let\eqld@colsepmax@val\eqld@box@colsep
6012   \let\eqld@box@colsep\eqld@break@col@sep}
6013 \eqld@define@key{equations,setup}{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
6014   {0,\eqld@decide@false}{\def\eqld@break@col@shortsep{0pt}}},%
6015   {\relax{\def\eqld@break@col@shortsep{#1}}}}
6016 \eqld@define@key{box,setup}{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
6017   {0,\eqld@decide@false}{\def\eqld@box@shortsep{0pt}}},%
6018   {\relax{\def\eqld@box@shortsep{#1}}}}
6019 \eqld@define@key{box,setup}{condsep}{\eqld@decide@select{#3}{#2}{#1}{%
6020   {0,\eqld@decide@false}{\def\eqld@box@condsep{0pt}}},%
6021   {{short,*}{\def\eqld@box@condsep{\eqld@box@shortsep}}},%
6022   {\relax{\def\eqld@box@condsep{#1}}}}
6023 \eqld@define@key{box,setup}{matrixsep}{\eqld@decide@select{#3}{#2}{#1}{%
6024   {0,\eqld@decide@false}{\def\eqld@box@matrixsep{0pt}}},%
6025   {{short,*}{\def\eqld@box@matrixsep{\eqld@box@shortsep}}},%
6026   {\relax{\def\eqld@box@matrixsep{#1}}}}

```

Horizontal Shape. Configure horizontal alignment schemes:

```

6027 \eqld@define@key\eqld@keyall{shape}[default]{\eqld@shape@set{#1}}
6028 \eqld@define@key\eqld@keyall{padding,pad}[indent]{%
6029   \eqld@decide@select{#3}{#2}{#1}{%
6030     {{max}{\let\eqld@paddingleft@val\@undefined}},%
6031     {{indent}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
6032     {{0,\eqld@decide@false}{\def\eqld@paddingleft@val{0pt}}},%
6033     {\relax{\def\eqld@paddingleft@val{#1}}}}
6034   \let\eqld@paddingright@val\eqld@paddingleft@val}
6035 \eqld@define@key\eqld@keyall{padleft}[indent]{%
6036   \eqld@decide@select{#3}{#2}{#1}{%
6037     {{max}{\let\eqld@paddingleft@val\@undefined}},%

```



```

6038    {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
6039    {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
6040    {\relax{\def\eql@paddingleft@val{#1}}}}
6041 \eql@define@key\eql@keyall{padright}[indent]{%
6042   \eql@decide@select{#3}{#2}{#1}}{%
6043    {{max}}{\let\eql@paddingright@val\@undefined}},%
6044    {{indent}}{\def\eql@paddingright@val{\eql@indent@val}}},%
6045    {{0,\eql@decide@false}}{\def\eql@paddingright@val{0pt}}},%
6046    {\relax{\def\eql@paddingright@val{#1}}}}
6047 \eql@define@key\eql@keyall{indent}[2em]{%
6048   \def\eql@indent@val{#1}}

```

TODO: describe

```

6049 \def\eql@shape@rel{\eqncontrol{align=left}}
6050 \def\eql@shape@cont{\eqncontrol{align=left,shift=*}}
6051 \eql@define@key\eql@keyall{shapereel}[]{\def\eql@shape@rel{#1}}
6052 \eql@define@key\eql@keyall{shapecont}[]{\def\eql@shape@cont{#1}}

```

TODO: describe

```

6053 \eql@define@key{control}{align}[]{%
6054   \eql@decide@select{#3}{#2}{#1}}{%
6055    {{l,left}}{\global\eql@append\eql@cell@container{\eql@shape@pos@z}},%
6056    {{c,center}}{\global\eql@append\eql@cell@container{\eql@shape@pos@one}},%
6057    {{r,right}}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw}}}}
6058 \eql@define@key{control}{shift,shifto}[]{%
6059   \eql@decide@select{#3}{#2}{#1}}{%
6060    {{*,indent}}{\eql@shape@alignamount@set{\eql@indent}},%
6061    {{!,outdent}}{\eql@shape@alignamount@set{-\eql@indent}},%
6062    {\relax{\eql@shape@alignamount@set{#1}}}}
6063 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

6064 \eql@define@key\eql@keyall{classout}{\eql@class@innerleft@set{#1}}
6065 \eql@define@key\eql@keyall{classin}{\eql@class@innerright@set{#1}}
6066 \eql@define@key\eql@keyall{classlead,classin*}{\eql@class@innerlead@set{#1}}
6067 \eql@define@key\eql@keyall{rel}{\def\eql@class@rel@symb{#1}}
6068 \eql@define@key\eql@keyall{classbreak}{\eql@class@rel@break@set{#1}}
6069 \eql@define@key\eql@keyall{classstart}{\eql@class@rel@start@set{#1}}
6070 \eql@define@key\eql@keyall{classbreakskip}[adjust]{%
6071   \eql@decide@select{#3}{#2}{#1}}{%
6072    {{0,\eql@decide@false}}{\def\eql@class@rel@break@skip{0mu}},%
6073    {{adjust,*}}{\def\eql@class@rel@break@skip{\thickmuskip-\medmuskip}},%
6074    {\relax{\def\eql@class@rel@break@skip{#1}}}}
6075 \eql@define@key{control}{rel}{\eql@class@rel@symb}{\eql@class@rel@make{#1}}
6076 \eql@define@key{control}{rel;}[]{\eql@class@rel@make{}}
6077 \eql@define@key{control}{rel*}[]{\eql@class@rel@make{}}
6078 \eql@define@key\eql@keyall{ampeq}[]{\eql@class@ampeq}
6079 \eql@define@key\eql@keyall{eqamp}[]{\eql@class@eqamp}
6080 \eql@define@key\eql@keyall{class}{\eql@decide@select{#3}{#2}{#1}}{%
6081   {{ampeq,amprel,eafter,beforerel}}{\eql@class@ampeq},%
6082   {{eqamp,relamp,eqbefore,afterrel}}{\eql@class@eqamp}}

```

Math Styles. Configure math classes at alignment marker:

```

6083 \eql@define@key\eql@keyall{style}[display]{%
6084   \eql@decide@select{#3}{#2}{#1}}{%

```



```

6085    {{text}}{\let\eq\@mathstyle\@empty}},%
6086    {{display}}{\let\eq\@mathstyle\displaystyle}}}}
6087 \eq\@define@key{setup}{casesstyle}[display]{%
6088   \eq\@decide@select{#3}{#2}{#1}{%
6089     {\eq\@decide@false{\let\eq\@cases\@mathstyle\eq\@false}},%
6090     {{text}}{\let\eq\@cases\@mathstyle\@empty}},%
6091     {{display}}{\let\eq\@cases\@mathstyle\displaystyle}}}}

```

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

```

6092 \def\eq\@punct@all#1#2#3#4#5\eq\@punct@end{%
6093   \def\eq\@tmp{#4}\def\eq\@tmpa{1}%
6094   \ifx\eq\@tmp\eq\@tmpa
6095     \ifnum#5=1111\relax
6096       \eq\@punct@set\eq\@punct@col{#1}%
6097       \eq\@punct@set\eq\@punct@line{#2}%
6098       \eq\@punct@set\eq\@punct@block{#3}%
6099     \else\ifnum#5=111\relax
6100       \eq\@punct@set\eq\@punct@line{#1}%
6101       \eq\@punct@set\eq\@punct@block{#2}%
6102     \else\ifnum#5=11\relax
6103       \eq\@punct@set\eq\@punct@block{#1}%
6104     \else
6105       \eq\@punct@clear
6106       \fi\fi\fi
6107   \else
6108     \eq\@error{Too many arguments to punctall}%
6109   \fi
6110 }

```

TODO: describe

```

6111 \eq\@define@key\eq\@keyall{punctsep}[\,]{\def\eq\@punct@sep{#1}}
6112 \eq\@define@key\eq\@keyall{punctclass}[\mathclose{}]{\def\eq\@punct@class{#1}}
6113 \eq\@define@key\eq\@keyall{punct}[.]{\eq\@punct@set\eq\@punct@block{#1}}
6114 \eq\@define@key\eq\@keyall{punct*}[]{\eq\@punct@set\eq\@punct@block\relax}
6115 \eq\@define@key\eq\@keyall{punctline}[,]{\eq\@punct@set\eq\@punct@line{#1}}
6116 \eq\@define@key\eq\@keyall{punctline*}[]{\eq\@punct@set\eq\@punct@line\relax}
6117 \eq\@define@key\eq\@keyall{punctcol}[,]{\eq\@punct@set\eq\@punct@col{#1}}
6118 \eq\@define@key\eq\@keyall{punctcol*}[]{\eq\@punct@set\eq\@punct@col\relax}
6119 \eq\@define@key{box}{punctcases}[,]{\eq\@punct@set\eq\@punct@col{#1}}
6120 \eq\@define@key{box}{punctcases*}[]{\eq\@punct@set\eq\@punct@col\relax}
6121 \eq\@define@key{setup}{punctcases}[,]{\eq\@punct@set\eq\@punct@cases{#1}}
6122 \eq\@define@key{setup}{punctcases*}[]{\eq\@punct@set\eq\@punct@cases\relax}
6123 \eq\@define@key\eq\@keyall{punctall}[,.,.]{\eq\@punct@all#111111\eq\@punct@end}
6124 \eq\@define@key{box}{punctterm}[true]{%
6125   \eq\@decide@bool{#3}{#2}{#1}\eq\@box@punct@term}

6126 \eq\@define@key{control}{punctsep}[\,]{\def\eq\@punct@sep{#1}}
6127 \eq\@define@key{control}{setpunct}[.]{\eq\@punct@set\eq\@punct@next{#1}}
6128 \eq\@define@key{control}{setpunct}[,]{\eq\@punct@set\eq\@punct@next{#1}}
6129 \eq\@define@key{control}{setpunct*}[]{\let\eq\@punct@next\relax}
6130 \eq\@define@key{control}{punct,punctapply}[\relax]{%
6131   \eq\@punct@set\eq\@punct@next{#1}\eq\@punct@apply@top}
6132 \eq\@define@key{control}{punctline}[,]{\eq\@punct@print@line}
6133 \eq\@define@key{control}{punctcol}[,]{\eq\@punct@print@col}

```

Frames. **TODO:** describe

```

6134 \eqld@define@key{box}{frame}[\fbox]{%
6135   \def\eqld@box@frame{#1}%
6136   \ifx\eqld@box@frame\@empty\let\eqld@box@frame\@firstofone\fi}
6137 \eqld@define@key{box}{wrap}{\eqld@box@wrap#1}
6138 \eqld@define@key{box}{delim}[r]{\eqld@decide@delim{#3}{#2}{#1}}
6139 \eqld@define@key{box}{ldelim}{\eqld@box@ldelim#1}
6140 \eqld@define@key{box}{rdelim}{\eqld@box@rdelim#1}
6141 \eqld@define@key{box}{lbrace}[]{\eqld@box@ldelim\lbrace}
6142 \eqld@define@key{box}{rbrace}[]{\eqld@box@rdelim\rbrace}
6143 \eqld@define@key{box}{lrbbrace,lrbaces}[]{\eqld@box@delim\lbrace\rbrace}
6144 \eqld@define@key{box}{braces}[lr]{%
6145   \eqld@decide@select{#3}{#2}{#1}{%
6146     {\eqld@decide@false}{\eqld@box@wrap}{}}},%
6147     {{l,left}{\eqld@box@ldelim\lbrace}}},%
6148     {{r,right}{\eqld@box@rdelim\rbrace}}},%
6149     {{\eqld@decide@true,lr,both}{\eqld@box@delim\lbrace\rbrace}}}}

```

TODO: describe

```

6150 \eqld@define@key{control}{framecell}[\fbox]{%
6151   \global\eqld@append\eqld@cell@container{\def\eqld@frame@cmd{#1}}}
6152 \eqld@define@key{control}{frametag}[\fbox]{%
6153   \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

```

6154 \eqld@define@key{equations,box}{alt}{}

```

Injections.

```

6155 \eqld@define@key{control}{inject}{%
6156   \global\eqld@append\eqld@interline@container{%
6157     \eqld@append\eqld@display@injectbefore{#1}}}
6158 \eqld@define@key{control}{inject*}{%
6159   \global\eqld@append\eqld@interline@container{%
6160     \eqld@append\eqld@display@injectafter{#1}}}
6161 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
6162 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
6163 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
6164 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}

```

TODO: describe

```

6165 \eqld@define@key{markline}{pos}{%
6166   \eqld@decide@select{#3}{#2}{#1}{%
6167     {{below,push}{\let\eqld@markline@pos\eqld@markline@pos@below}}},%
6168     {{baseline}{\let\eqld@markline@pos\eqld@markline@pos@baseline}}},%
6169     {{bottom}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}
6170 \eqld@define@key{markline}{below,push}[]{%
6171   \let\eqld@markline@pos\eqld@markline@pos@below}
6172 \eqld@define@key{markline}{baseline}[]{%
6173   \let\eqld@markline@pos\eqld@markline@pos@baseline}
6174 \eqld@define@key{markline}{bottom}[]{%
6175   \let\eqld@markline@pos\eqld@markline@pos@bottom}
6176 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
6177 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}
6178 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
6179 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}

```

```

6180 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
6181 \eqld@define@key{setup}{markpos}{%
6182   \eqld@decide@select{#3}{#2}{#1}{%
6183     {{below}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
6184     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
6185     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}

```

Global Switches. Set global switches:

```

6186 \let\eqld@multi@linesfallback\eqld@false
6187 \let\eqld@scan@par\eqld@false
6188 \let\eqld@single@cr@mode\eqld@false
6189 \let\eqld@amp@mode\eqld@true
6190 \let\eqld@ampproof@active\eqld@false
6191 \let\eqld@parseopt@warn@env\eqld@warn@parseopt
6192 \let\eqld@parseopt@warn@cr\empty

6193 \eqld@define@key{equations,setup}{linesfallback}[true]{%
6194   \eqld@decide@select{#3}{#2}{#1}{%
6195     {\eqld@decide@false{\let\eqld@multi@linesfallback\eqld@false}},%
6196     {{reuse,lean}}{\let\eqld@multi@linesfallback\z@}},%
6197     {{measure,full,\eqld@decide@true}}{\let\eqld@multi@linesfallback\eqld@true}}}}
6198 \eqld@define@key{setup}{ampproof}[true]{%
6199   \eqld@decide@bool{#3}{#2}{#1}\eqld@ampproof@active}
6200 \eqld@define@key{equations,setup}{equationcr}[true]{%
6201   \eqld@decide@select{#3}{#2}{#1}{%
6202     {\eqld@decide@false{\let\eqld@single@cr@mode\eqld@false}},%
6203     {{\eqld@decide@true,break}}{\let\eqld@single@cr@mode\eqld@break@cr}},%
6204     {{error,verbose}}{\let\eqld@single@cr@mode\eqld@single@cr@error}}}}
6205 \eqld@define@key\eqld@keyall{amp}[true]{%
6206   \eqld@decide@bool{#3}{#2}{#1}\eqld@amp@mode}}
6207 \eqld@define@key\eqld@keyall{rescan}[true]{%
6208   \eqld@decide@if{#3}{#2}{#1}{%
6209     {\let\eqld@scan@body\eqld@scan@body@rescan}%
6210     {\let\eqld@scan@body\eqld@scan@body@dump}}}
6211 \eqld@define@key\eqld@keyall{scanpar}[true]{%
6212   \eqld@decide@bool{#3}{#2}{#1}\eqld@scan@par}
6213 \eqld@define@key{setup}{defaults}{%
6214   \eqld@decide@select{#3}{#2}{#1}{%
6215     {{classic}}{\eqld@defaults@classic}},%
6216     {{eqnlines}}{\eqld@defaults@eqnlines}}}}
6217 \eqld@define@key\eqld@keyall{verbose}[true]{%
6218   \eqld@decide@if{#3}{#2}{#1}\eqld@verbose@on\eqld@verbose@off}

```

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

```

6219 \let\eqld@provide@opt@env\tw@
6220 \let\eqld@provide@opt@amsmathpatch\eqld@false
6221 \let\eqld@provide@opt@backup\eqld@false
6222 \let\eqld@provide@opt@ang\eqld@true
6223 \let\eqld@provide@opt@eqref\eqld@true
6224 \let\eqld@provide@opt@matrix\eqld@true

6225 \eqld@define@key{setup}{amsmathends,amsmathpatch}[true]{%
6226   \eqld@error@packageoption{#2}%
6227   \eqld@decide@bool{#3}{#2}{#1}\eqld@provide@opt@amsmathpatch}
6228 \eqld@define@key{setup}{backup}[true]{%
6229   \eqld@error@packageoption{#2}%

```

```

6230 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
6231 \eql@define@key{setup}{env}[equation]{%
6232 \eql@error@packageoption{#2}%
6233 \eql@decide@select{#3}{#2}{#1}{%
6234 {none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
6235 {equation,latex}{\let\eql@provide@opt@env\one}},%
6236 {amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}%
6237 \eql@define@key{setup}{ang}[true]{%
6238 \eql@error@packageoption{#2}%
6239 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
6240 \eql@define@key{setup}{eqref}[true]{%
6241 \eql@error@packageoption{#2}%
6242 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}
6243 \eql@define@key{setup}{matrix}[true]{%
6244 \eql@error@packageoption{#2}%
6245 \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@matrix}

```

Shortcut Options. **TODO:** describe

```

6246 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
6247 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
6248 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
6249 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
6250 \def\eql@parseopt@matrix#1{\eqnaddopt{matrix}\eql@parseopt@peek}
6251 \def\eql@parseopt@cases#1{\eqnaddopt{cases}\eql@parseopt@peek}
6252 \def\eql@parseopt@shape#1#2{\eqnaddopt{shape={#2}}\eql@parseopt@peek}
6253 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
6254 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
6255 \def\eql@parseopt@eqamp#1{\eqnaddopt{eqamp}\eql@parseopt@peek}
6256 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
6257 \def\eql@parseopt@par#1{\eqnaddopt{par}\eql@parseopt@peek}
6258 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
6259 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
6260 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
6261 \def\eql@parseopt@punctclear#1{\eql@parseopt@peek'~}
6262 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
6263 \def\eql@parseopt@punctnext#1#2{%
6264 \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
6265 \def\eql@parseopt@punctblock#1#2{%
6266 \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
6267 \def\eql@parseopt@punctterm#1{\let\eql@punct@term\eql@true\eql@parseopt@peek}
6268 \def\eql@parseopt@rel symb#1{\eql@parseopt@peek?\eql@class@rel@symb}
6269 \def\eql@parseopt@rel cont#1{\eql@parseopt@peek?{}}
6270 \def\eql@parseopt@rel start#1{\eql@parseopt@peek?\relax}
6271 \def\eql@parseopt@rel ord#1{\eql@parseopt@peek?{}}
6272 \def\eql@parseopt@vspace[#1]{%
6273 \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:

```

6274 \def\eql@defaults@classic{%
6275   \eqnlineset{numberline=all}%
6276   \eqnlineset{condsep={1em}}%
6277   \eqnlineset{matrixsep={2\arraycolsep}}%
6278   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
6279   \eqnlineset{maxcolsep=off}%
6280   \eqnlineset{spread={\jot}}%
6281   \eqnlineset{tagmargin}%
6282   \eqnlineset{tagmarginratio=1}%
6283   \eqnlineset{tagmarginthreshold=0.5}%
6284   \eqnlineset{leftmargin={\leftmargini}}%
6285   \eqnlineset{padding=max}%
6286   \eqnlineset{evadetag=off}%
6287   \eqnlineset{displayheight=off}%
6288   \eqnlineset{displaydepth=off}%
6289   \eqnlineset{shortmode=belowsingle}%
6290   \eqnlineset{abovecontmode=short}%
6291   \eqnlineset{belowcontmode=short}%
6292   \eqnlineset{aboveparmode=long}%
6293   \eqnlineset{belowparmode=long}%
6294   \eqnlineset{abovetopmode=long}%
6295   \eqnlineset{belowtopmode=long}%
6296   \eqnlineset{abovelongskip={\abovedisplayskip}}%
6297   \eqnlineset{belowlongskip={\belowdisplayskip}}%
6298   \eqnlineset{aboveshortskip={\abovedisplayskip}}%
6299   \eqnlineset{belowshortskip={\belowdisplayskip}}%
6300   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
6301   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
6302   \eqnlineset{abovecontskip=0pt}%
6303   \eqnlineset{belowcontskip=0pt}%
6304   \eqnlineset{aboveparskip=0pt}%
6305   \eqnlineset{belowparskip=0pt}%
6306   \eqnlineset{abovetopskip=0pt}%
6307   \eqnlineset{belowtopskip=0pt}%
6308   \eqnlineset{abovetagskip=0pt}%
6309   \eqnlineset{belowtagskip=0pt}%
6310   \eqnlineset{allowbreaks=0}%
6311   \eqnlineset{equationcr=off}%
6312   \eqnlineset{amp=off}%
6313   \eqnlineset{modifier=off}%
6314   \eqnlineset{linesfallback=false}%
6315   \eqnlineset{casesstyle=text}%
6316   \eqnlineset{sgropt=nonumber}%
6317   \eqnlineset{angopt=nonumber}%
6318 }

```

values based on 10pt vs 12pt

eql@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):

```

6319 \def\eql@defaults@eqnlines{%
6320   \eqnlineset{numberline=all}%
6321   \eqnlineset{condsep=short}%
6322   \eqnlineset{matrixsep=short}%

```

```

6323 \eqnlineset{mintagsep=.5em}%
6324 \eqnlineset{maxcolsep=2em}%
6325 \eqnlineset{spread={0.2\normalbaselineskip}}%
6326 \eqnlineset{tagmargin}%
6327 \eqnlineset{tagmarginratio=.334}%
6328 \eqnlineset{tagmarginthreshold=0.5}%
6329 \eqnlineset{leftmargin={\leftmargini}}%
6330 \eqnlineset{padding=0pt}%
6331 \eqnlineset{evadetag}%
6332 \eqnlineset{displayheight=strut}%
6333 \eqnlineset{displaydepth=strut}%
6334 \eqnlineset{shortmode=above}%
6335 \eqnlineset{abovecontmode=noskip}%
6336 \eqnlineset{belowcontmode=long}%
6337 \eqnlineset{aboveparmode=long}%
6338 \eqnlineset{belowparmode=long}%
6339 \eqnlineset{abovetopmode=noskip}%
6340 \eqnlineset{belowtopmode=long}%
6341 \eqnlineset{longskip={0.75\normalbaselineskip
6342   plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
6343 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
6344   plus 0.25\normalbaselineskip}}%
6345 \eqnlineset{belowshortskip={0.0\normalbaselineskip
6346   plus 0.25\normalbaselineskip}}%
6347 \eqnlineset{medskip={0.4\normalbaselineskip
6348   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
6349 \eqnlineset{abovecontskip=0pt}%
6350 \eqnlineset{belowcontskip=0pt}%
6351 \eqnlineset{aboveparskip=0pt}%
6352 \eqnlineset{belowparskip=0pt}%
6353 \eqnlineset{abovetopskip=0pt}%
6354 \eqnlineset{belowtopskip=0pt}%
6355 \eqnlineset{abovetagskip={0.25\normalbaselineskip
6356   minus 0.25\normalbaselineskip}}%
6357 \eqnlineset{belowtagskip={0.25\normalbaselineskip
6358   minus 0.25\normalbaselineskip}}%
6359 \eqnlineset{allowbreaks=3}%
6360 \eqnlineset{equationcr=break}%
6361 \eqnlineset{amp=on}%
6362 \eqnlineset{modifier=on,modifierenv=off}%
6363 \eqnlineset{linesfallback=true}%
6364 \eqnlineset{casesstyle=false}%
6365 \eqnlineset{sgropt}%
6366 \eqnlineset{angopt}%
6367 }

```

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.

`\eq@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```

\eq@provide@moveenv
@provide@undefinecmd
@provide@undefineenv
6368 \def\eq@provide@movecmd#1#2{%
6369   \eq@letcs{#1\expandafter}\csname#2\endcsname

```

```

6370 }
6371 \def\eql@provide@moveenv#1#2{%
6372   \eql@provide@movecmd{#1}{#2}%
6373   \ifcsname end#2\endcsname
6374     \eql@provide@movecmd{end#1}{end#2}%
6375   \fi
6376 }
6377 \def\eql@provide@undefinecmd#1{%
6378   \eql@letcs{#1}\@undefined
6379 }
6380 \def\eql@provide@undefineenv#1{%
6381   \eql@provide@undefinecmd{#1}%
6382   \eql@provide@undefinecmd{end#1}%
6383 }

```

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:

```

6384 \def\eql@amsmath@endfix#1#2{%
6385   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
6386   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
6387     \eql@provide@movecmd{end#1}{end#2}%
6388   \fi
6389 }

```

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

`amsmath@fixmatrixend`

```

6390 \def\eql@amsmath@fixmatrix#1{%
6391   \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
6392   \begingroup
6393     \let\matrix@check\@gobble
6394     \def\env@matrix{\noexpand\env@matrix}%
6395     \def\env@cases{\noexpand\env@cases}%
6396     \global\edef\eql@tmp{\eql@tmp}%
6397   \endgroup
6398   \eql@letcs{#1}\eql@tmp
6399 }
6400 \def\eql@amsmath@fixmatrixend#1{%
6401   \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
6402   \begingroup
6403     \expandafter\def\expandafter\endmatrix\expandafter{%
6404       \expandafter\noexpand\expandafter{\endmatrix}}%
6405     \global\long\edef\eql@tmp{\eql@tmp}%
6406   \endgroup
6407   \eql@letcs{end#1}\eql@tmp
6408 }

```

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

```

6409 \def\eql@amsmath@fixends{%
6410   \eql@amsmath@after{%
6411     \eql@amsmath@endfix{flalign}{align}%
6412     \eql@amsmath@endfix{alignat}{align}%
6413     \eql@amsmath@endfix{xalignat}{align}%
6414     \eql@amsmath@endfix{xxalignat}{align}%
6415     \eql@amsmath@endfix{gather*}{gather}%
6416     \eql@amsmath@endfix{multline*}{multline}%
6417     \eql@amsmath@endfix{align*}{align}%
6418     \eql@amsmath@endfix{flalign*}{align}%
6419     \eql@amsmath@endfix{alignat*}{align}%
6420     \eql@amsmath@endfix{xalignat*}{align}%
6421     \eql@amsmath@endfix{gathered}{aligned}%
6422     \eql@amsmath@endfix{alignedat}{aligned}%
6423   }
6424 }

```

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

```

6425 \def\eql@amsmath@fixmatrices{%
6426   \eql@amsmath@after{%
6427     \eql@amsmath@fixmatrix{cases}%
6428     \eql@amsmath@fixmatrix{matrix}%
6429     \eql@amsmath@fixmatrix{pmatrix}%
6430     \eql@amsmath@fixmatrixend{pmatrix}%
6431     \eql@amsmath@fixmatrixend{bmatrix}%
6432     \eql@amsmath@fixmatrixend{Bmatrix}%
6433     \eql@amsmath@fixmatrixend{vmatrix}%
6434     \eql@amsmath@fixmatrixend{Vmatrix}%
6435   }
6436 }

```

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

```

6437 \def\eql@provide@backup@amsenv#1{%
6438   \eql@amsmath@after{%
6439     \eql@provide@moveenv{ams#1}{#1}%
6440     \eql@tagging@register@luamml{ams#1}%
6441     \eql@markline@amsthm@move{ams#1}{#1}%
6442   }%
6443 }

```

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

```

6444 \def\eql@provide@backup@amsbox#1{%
6445   \eql@amsmath@after{%
6446     \eql@provide@moveenv{ams#1}{#1}%
6447   }%
6448 }

```

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

```

6449 \def\eql@provide@backup@eqref{%
6450   \eql@amsmath@after{%

```



```

6451 \eql@provide@movecmd{amseqref}{eqref}%
6452 }%
6453 }

```

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

```

6454 \def\eql@provide@backup@multlined{%
6455 \AddToHook{package/mathtools/after}{%
6456 \eql@provide@moveenv{amsmultlined}{multlined}}%
6457 }

```

`vide@backup@equation` The \LaTeX environment `equation` is overwritten by several packages to implement their adjustments. Here we cater for adjustments through `amsmath`, `hyperref` and the PDF tagging mechanism. Copy `equation` and `equation*` whenever `amsmath` is loaded. Whenever `hyperref` is loaded, and `amsmath` is not yet present, backup the original \LaTeX and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original \LaTeX `equation`. The PDF tagging mechanism registers `equation` upon `\begin{document}`. We thus need to register all copies of `equation` on our own, so that they can be used with their new names:

```

6458 \def\eql@provide@backup@equation{%
6459 \eql@amsmath@after{%
6460 \eql@provide@moveenv{amsequation}{equation}%
6461 \eql@provide@moveenv{amsequation*}{equation*}%
6462 \eql@tagging@register@env{amsequation}%
6463 \eql@tagging@register@env{amsequation*}%
6464 \eql@tagging@register@luamml{amsequation}%
6465 \eql@tagging@register@luamml{amsequation*}%
6466 \eql@markline@amsthm@move{amsequation}{equation}%
6467 \eql@markline@amsthm@move{amsequation*}{equation*}%
6468 }%
6469 \AddToHook{package/hyperref/after}{%
6470 \@ifpackageloaded{amsmath}{}%
6471 \eql@provide@moveenv{hyperrefequation}{equation}%
6472 \eql@tagging@register@env{hyperrefequation}%
6473 \eql@tagging@register@luamml{hyperrefequation}%
6474 \eql@markline@amsthm@move{hyperrefequation}{equation}%
6475 }%
6476 }%
6477 \@ifpackageloaded{amsmath}{}%
6478 \@ifpackageloaded{hyperref}{%
6479 \let\latexequation\H@equation
6480 \let\endlatexequation\H@endequation
6481 }{\eql@provide@moveenv{latexequation}{equation}}%
6482 \eql@tagging@register@env{latexequation}%
6483 \eql@tagging@register@luamml{latexequation}%
6484 \eql@markline@amsthm@move{latexequation}{equation}%
6485 }%
6486 }

```

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

```

6487 \def\eql@provide@backup@displaymath{%
6488 \eql@provide@moveenv{latexdisplaymath}{displaymath}%
6489 \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
6490 }

```

`@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:

```

6491 \def\eql@provide@backup@subequations{%
6492   \eql@amsmath@after{%
6493     \eql@provide@moveenv{amssubequations}{subequations}%
6494   }%
6495   \AddToHook{package/hyperref/after}{%
6496     \AddToHook{cmd/amssubequations/before}{%
6497       {%
6498         \stepcounter{equation}%
6499         \protected@edef\theHparentequation{\theHequation}%
6500         \addtocounter{equation}{-1}%
6501       }%
6502     \AddToHook{cmd/amssubequations/after}{%
6503       {%
6504         \def\theHequation{\theHparentequation\alph{equation}}%
6505         \ignorespaces
6506       }%
6507     }%
6508 }

```

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

```

6509 \def\eql@provide@backup{%
6510   \eql@provide@backup@eqref
6511   \eql@provide@backup@equation
6512   \eql@provide@backup@displaymath
6513   \eql@provide@backup@amsenv{gather}%
6514   \eql@provide@backup@amsenv{multline}%
6515   \eql@provide@backup@amsenv{align}%
6516   \eql@provide@backup@amsenv{flalign}%
6517   \eql@provide@backup@amsenv{alignat}%
6518   \eql@provide@backup@amsenv{xalignat}%
6519   \eql@provide@backup@amsenv{xxalignat}%
6520   \eql@provide@backup@amsenv{gather*}%
6521   \eql@provide@backup@amsenv{multline*}%
6522   \eql@provide@backup@amsenv{align*}%
6523   \eql@provide@backup@amsenv{flalign*}%
6524   \eql@provide@backup@amsenv{alignat*}%
6525   \eql@provide@backup@amsenv{xalignat*}%
6526   \eql@provide@backup@amsbox{gathered}%
6527   \eql@provide@backup@multlined
6528   \eql@provide@backup@amsbox{aligned}%
6529   \eql@provide@backup@amsbox{alignedat}%
6530   \eql@provide@backup@amsbox{cases}%
6531   \eql@provide@backup@amsbox{matrix}%
6532   \eql@provide@backup@amsbox{pmatrix}%
6533   \eql@provide@backup@amsbox{bmatrix}%
6534   \eql@provide@backup@amsbox{Bmatrix}%
6535   \eql@provide@backup@amsbox{vmatrix}%
6536   \eql@provide@backup@amsbox{Vmatrix}%
6537   \eql@provide@backup@subequations
6538 }

```

Replacement `amsmath` Environments. **TODO:** describe

```

6539 \def\eql@alignat@gobblecol#1{%
6540   \eql@ifnextchar@tight\bgroup{\@firstoftwo{#1}}{#1}}

```

`eql@gathered` (*env.*) Define replacement versions for boxed environments `gathered`, `multlined` and `aligned`
`eql@multlined` (*env.*) which forward to `equationsbox` with specific presets:

```

eql@aligned (env.)
6541 \newenvironment{eql@gathered}%
6542   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
6543 \newenvironment{eql@multlined}%
6544   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
6545 \newenvironment{eql@aligned}%
6546   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
6547 \newenvironment{eql@alignedat}%
6548   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}%
6549   {\endequationsbox}
6550 \newenvironment{eql@cases}%
6551   {\eqnaddopt{cases}\equationsbox}{\endequationsbox}
6552 \newenvironment{eql@matrix}%
6553   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=.}\equationsbox}%
6554   {\endequationsbox}
6555 \newenvironment{eql@pmatrix}%
6556   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=r}\equationsbox}%
6557   {\endequationsbox}
6558 \newenvironment{eql@bmatrix}%
6559   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=s}\equationsbox}%
6560   {\endequationsbox}
6561 \newenvironment{eql@Bmatrix}%
6562   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=c}\equationsbox}%
6563   {\endequationsbox}
6564 \newenvironment{eql@vmatrix}%
6565   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=v}\equationsbox}%
6566   {\endequationsbox}
6567 \newenvironment{eql@Vmatrix}%
6568   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=d}\equationsbox}%
6569   {\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.*)

```

6570 \newenvironment{eql@equation}%
6571   {\eqnaddopt{equation,donumber}\equations}{\endequations}
6572 \newenvironment{eql@equation*}%
6573   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
6574 \newenvironment{eql@displaymath}%
6575   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
6576 \newenvironment{eql@gather}%
6577   {\eqnaddopt{lines,donumber}\equations}{\endequations}
6578 \newenvironment{eql@gather*}%
6579   {\eqnaddopt{lines,nonumber}\equations}{\endequations}
6580 \newenvironment{eql@multline}%
6581   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,donumber}%
6582     \equations}{\endequations}
6583 \newenvironment{eql@multline*}%
6584   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,nonumber}%
6585     \equations}{\endequations}
6586 \newenvironment{eql@align}%
6587   {\eqnaddopt{columns,donumber}\equations}{\endequations}
6588 \newenvironment{eql@align*}%

```

```

6589 {\eqnadopt{columns,nonumber}\equations}{\endequations}
6590 \newenvironment{eql@flalign}%
6591 {\eqnadopt{columns,fulllength,donumber}\equations}{\endequations}
6592 \newenvironment{eql@flalign*}%
6593 {\eqnadopt{columns,fulllength,nonumber}\equations}{\endequations}
6594 \newenvironment{eql@alignat}%
6595 {\eqnadopt{columns,colsep=off,donumber}%
6596 \eql@alignat@gobblecol\equations}{\endequations}
6597 \newenvironment{eql@xalignat}%
6598 {\eqnadopt{columns,donumber}%
6599 \eql@alignat@gobblecol\equations}{\endequations}
6600 \newenvironment{eql@xxalignat}%
6601 {\eqnadopt{columns,fulllength,donumber}%
6602 \eql@alignat@gobblecol\equations}{\endequations}
6603 \newenvironment{eql@alignat*}%
6604 {\eqnadopt{columns,colsep=off,nonumber}%
6605 \eql@alignat@gobblecol\equations}{\endequations}
6606 \newenvironment{eql@xalignat*}%
6607 {\eqnadopt{columns,nonumber}%
6608 \eql@alignat@gobblecol\equations}{\endequations}

```

Install Additional Environments. The additional environments need to be installed at their intended names which can be adjusted by the user.

`\eql@provide@onlyonce` Process arguments for providing a specific environment. #1 describes the environment using the `amsmath` name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the `amsmath` environment in place making sure that the replacement is robust against loading `amsmath` before or after. If #2 equals ‘*’, just overwrite the `amsmath` environment in place immediately (e.g. within a block in the document body):

```

6609 \def\eql@provide@onlyonce#1#2{%
6610 \def\eql@tmp{#2}\def\eql@tmpa{#1}%
6611 \ifx\eql@tmp\eql@tmpa
6612 \let\eql@tmp\@empty
6613 \fi
6614 \ifx\eql@tmp\@empty
6615 \let\eql@tmp\@undefined
6616 \ifx\@nodocument\relax
6617 \def\eql@tmp{#1}%
6618 \fi
6619 \ifcsname eql@provided@#1\endcsname
6620 \def\eql@tmp{#1}%
6621 \fi
6622 \eql@letcs{eql@provided@#1}\eql@true
6623 \else
6624 \def\eql@tmpa{*}%
6625 \ifx\eql@tmp\eql@tmpa
6626 \def\eql@tmp{#1}%
6627 \fi
6628 \fi
6629 }

```

`\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:

```

6630 \def\eql@provide@eqref#1{%
6631   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
6632   \ifx\eql@tmp\eql@tmpa
6633     \let\eql@tmp\@empty
6634   \fi
6635   \ifx\eql@tmp\@empty
6636     \eql@provide@onlyonce{eqref}{}%
6637   \else
6638     \def\eql@tmpa{*}%
6639     \ifx\eql@tmp\eql@tmpa
6640       \def\eql@tmp{eqref}%
6641     \else
6642       \edef\eql@tmp{\expandafter\@gobble\string#1}%
6643     \fi
6644   \fi
6645   \ifdefined\eql@tmp
6646     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
6647   \else
6648     \eql@amsmath@after{%
6649       \eql@provide@movecmd{eqref}{eql@eqref}%
6650     }%
6651     \AddToHook{package/mathtools/after}{%
6652       \eql@provide@movecmd{eqref}{eql@eqref}%
6653     }%
6654     \eql@provide@movecmd{eqref}{eql@eqref}%
6655     \eql@amsmath@undefine\eqref
6656   \fi
6657 }

```

`\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:

```

6658 \def\eql@provide@amsenv#1#2{%
6659   \eql@provide@onlyonce{#1}{#2}%
6660   \ifdefined\eql@tmp
6661     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6662     \eql@tagging@register@luamml{\eql@tmp}%
6663     \eql@markline@amsthm@register{\eql@tmp}%
6664   \else
6665     \eql@amsmath@after{%
6666       \eql@provide@moveenv{#1}{eql@#1}%
6667       \eql@markline@amsthm@register{#1}%
6668     }%
6669     \AddToHook{package/mathtools/after}{%
6670       \eql@provide@moveenv{#1}{eql@#1}%
6671       \eql@markline@amsthm@register{#1}%
6672     }%
6673     \eql@provide@moveenv{#1}{eql@#1}%
6674     \eql@markline@amsthm@register{#1}%
6675     \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6676   \fi
6677 }

```

`\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:

```

6678 \def\eql@provide@amsbox#1#2{%
6679   \eql@provide@onlyonce{#1}{#2}%

```

```

6680 \ifdefined\eql@tmp
6681   \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6682 \else
6683   \eql@amsmath@after{%
6684     \eql@provide@moveenv{#1}{eql@#1}}%
6685   \AddToHook{package/mathtools/after}{%
6686     \eql@provide@moveenv{#1}{eql@#1}}%
6687   \eql@provide@moveenv{#1}{eql@#1}%
6688   \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6689 \fi
6690 }

```

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:

```

6691 \def\eql@provide@multlined#1{%
6692   \eql@provide@onlyonce{multlined}{#1}%
6693   \ifdefined\eql@tmp
6694     \eql@provide@moveenv{\eql@tmp}{eql@multlined}%
6695   \else
6696     \AddToHook{package/mathtools/after}{%
6697       \eql@provide@moveenv{multlined}{eql@multlined}}%
6698     \eql@provide@moveenv{multlined}{eql@multlined}%
6699     \@ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
6700       \eql@provide@undefineenv{multlined}}}%
6701   \fi
6702 }

```

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

```

6703 \def\eql@provide@matrix#1#2#3{%
6704   \eql@provide@onlyonce{#1}{#3}%
6705   \ifdefined\eql@tmp
6706     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6707     \eql@tagging@register@luamml{\eql@tmp}%
6708   \else
6709     \eql@amsmath@after{%
6710       \eql@provide@moveenv{#1}{eql@#1}%
6711     }%
6712     \eql@provide@moveenv{#1}{eql@#1}%
6713     \ifdefined#2\eql@amsmath@before{\eql@provide@undefineenv{#1}}\fi%
6714   \fi
6715 }

```

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:

```

6716 \def\eql@provide@equation#1{%
6717   \eql@provide@onlyonce{equation}{#1}%
6718   \ifdefined\eql@tmp
6719     \eql@provide@moveenv{\eql@tmp}{eql@equation}%
6720     \eql@tagging@register@luamml{\eql@tmp}%
6721     \eql@markline@amsthm@register{\eql@tmp}%
6722   \else
6723     \eql@amsmath@after{%

```

```

6724     \eql@provide@moveenv{equation}{eql@equation}%
6725     \eql@markline@amsthm@register{equation}%
6726 }%
6727 \AddToHook{package/hyperref/after}{%
6728     \ifpackageloaded{amsmath}{}{%
6729         \eql@provide@moveenv{equation}{eql@equation}%
6730         \eql@markline@amsthm@register{equation}%
6731     }%
6732 }%
6733 \eql@provide@moveenv{equation}{eql@equation}%
6734 \eql@markline@amsthm@register{equation}%
6735 \ifdefined\eql@tagging@on
6736     \AddToHook{begindocument/end}{%
6737         \eql@provide@moveenv{equation}{eql@equation}%
6738         \eql@markline@amsthm@register{equation}%
6739     }%
6740 \fi
6741 \fi
6742 }

```

`\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:

```

6743 \def\eql@provide@equationstar#1{%
6744     \eql@provide@onlyonce{equation*}{#1}%
6745     \ifdefined\eql@tmp
6746         \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
6747         \eql@tagging@register@luamml{\eql@tmp}%
6748         \eql@markline@amsthm@register{\eql@tmp}%
6749     \else
6750         \eql@amsmath@after{%
6751             \eql@provide@moveenv{equation*}{eql@equation*}%
6752             \eql@markline@amsthm@register{equation*}%
6753         }%
6754         \eql@provide@moveenv{equation*}{eql@equation*}%
6755         \eql@markline@amsthm@register{equation*}%
6756         \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
6757         \ifdefined\eql@tagging@on
6758             \AddToHook{begindocument/end}{%
6759                 \eql@provide@moveenv{equation*}{eql@equation*}%
6760                 \eql@markline@amsthm@register{equation*}%
6761             }%
6762         \fi
6763     \fi
6764 }

```

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

```

6765 \def\eql@provide@displaymath#1{%
6766     \eql@provide@onlyonce{displaymath}{#1}%
6767     \ifdefined\eql@tmp
6768         \eql@provide@moveenv{\eql@tmp}{eql@displaymath}%
6769         \eql@markline@amsthm@register{\eql@tmp}%
6770         \eql@tagging@register@luamml{\eql@tmp}%
6771     \else
6772         \eql@provide@moveenv{displaymath}{eql@displaymath}%
6773         \eql@markline@amsthm@register{displaymath}%

```

```

6774 \ifdefined\eql@tagging@on
6775 \AddToHook{begindocument/end}{%
6776 \eql@provide@moveenv{displaymath}{eql@displaymath}}%
6777 \fi
6778 \fi
6779 }

```

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:

```

6780 \def\eql@provide@subequations#1{%
6781 \eql@provide@onlyonce{subequations}{#1}%
6782 \ifdefined\eql@tmp
6783 \eql@provide@moveenv{\eql@tmp}{eql@subequations}%
6784 \else
6785 \eql@amsmath@after{%
6786 \eql@provide@moveenv{subequations}{eql@subequations}%
6787 }%
6788 \AddToHook{package/hyperref/after}{%
6789 \AddToHook{cmd/subequations/before}[hyperref]{}%
6790 \AddToHook{cmd/subequations/after}[hyperref]{}%
6791 \RemoveFromHook{cmd/subequations/before}[hyperref]%
6792 \RemoveFromHook{cmd/subequations/after}[hyperref]%
6793 \AddToHook{begindocument/end}{%
6794 \eql@provide@moveenv{subequations}{eql@subequations}}%
6795 }%
6796 \eql@provide@moveenv{subequations}{eql@subequations}%
6797 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6798 \fi
6799 }

```

\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:

```

6800 \def\eql@provide@sqr{%
6801 \let\[ \eql@sqr@open
6802 \let\] \eql@sqr@close
6803 \eql@amsmath@after{%
6804 \let\[ \eql@sqr@open
6805 \let\] \eql@sqr@close
6806 }%
6807 \ifdefined\eql@tagging@on
6808 \AddToHook{begindocument/end}{%
6809 \let\[ \eql@sqr@open
6810 \let\] \eql@sqr@close
6811 }%
6812 \fi
6813 }

```

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

```

6814 \def\eql@provide@ang{%

```



```

6815 \let<\eq@ang@open
6816 \let>\eq@ang@close
6817 }

```

TODO: describe

```

6818 \def\eq@provide@tagform{%
6819 \def\tagform##1{\maketag@@{\eq@tags@tagform{##1}}}}
6820 \def\eq@provide@maketag{%
6821 \def\maketag@@##1{\eq@tags@taglayout{##1}}}

```

Interface.

provide (*key*) We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

6822 \eq@define@key{provide}{equation}[]{\eq@provide@equation{#1}}
6823 \eq@define@key{provide}{equation*}[]{\eq@provide@equationstar{#1}}
6824 \eq@define@key{provide}{displaymath}[]{\eq@provide@displaymath{#1}}
6825 \eq@define@key{provide}{gather}[]{\eq@provide@amsenv{gather}{#1}}
6826 \eq@define@key{provide}{multline}[]{\eq@provide@amsenv{multline}{#1}}
6827 \eq@define@key{provide}{align}[]{\eq@provide@amsenv{align}{#1}}
6828 \eq@define@key{provide}{flalign}[]{\eq@provide@amsenv{flalign}{#1}}
6829 \eq@define@key{provide}{alignat}[]{\eq@provide@amsenv{alignat}{#1}}
6830 \eq@define@key{provide}{xalignat}[]{\eq@provide@amsenv{xalignat}{#1}}
6831 \eq@define@key{provide}{xxalignat}[]{\eq@provide@amsenv{xxalignat}{#1}}
6832 \eq@define@key{provide}{gather*}[]{\eq@provide@amsenv{gather*}{#1}}
6833 \eq@define@key{provide}{multline*}[]{\eq@provide@amsenv{multline*}{#1}}
6834 \eq@define@key{provide}{align*}[]{\eq@provide@amsenv{align*}{#1}}
6835 \eq@define@key{provide}{flalign*}[]{\eq@provide@amsenv{flalign*}{#1}}
6836 \eq@define@key{provide}{alignat*}[]{\eq@provide@amsenv{alignat*}{#1}}
6837 \eq@define@key{provide}{xalignat*}[]{\eq@provide@amsenv{xalignat*}{#1}}
6838 \eq@define@key{provide}{gathered}[]{\eq@provide@amsbox{gathered}{#1}}
6839 \eq@define@key{provide}{multlined}[]{\eq@provide@multlined{#1}}
6840 \eq@define@key{provide}{aligned}[]{\eq@provide@amsbox{aligned}{#1}}
6841 \eq@define@key{provide}{alignedat}[]{\eq@provide@amsbox{alignedat}{#1}}
6842 \eq@define@key{provide}{cases}[]{\eq@provide@matrix{cases}\eq@false{#1}}
6843 \eq@define@key{provide}{matrix}[]{\eq@provide@matrix{matrix}\eq@false{#1}}
6844 \eq@define@key{provide}{pmatrix}[]{\eq@provide@matrix{pmatrix}\eq@false{#1}}
6845 \eq@define@key{provide}{bmatrix}[]{\eq@provide@matrix{bmatrix}\eq@true{#1}}
6846 \eq@define@key{provide}{Bmatrix}[]{\eq@provide@matrix{Bmatrix}\eq@true{#1}}
6847 \eq@define@key{provide}{vmatrix}[]{\eq@provide@matrix{vmatrix}\eq@true{#1}}
6848 \eq@define@key{provide}{Vmatrix}[]{\eq@provide@matrix{Vmatrix}\eq@true{#1}}
6849 \eq@define@key{provide}{subequations}[]{\eq@provide@subequations{#1}}
6850 \eq@define@key{provide}{sqr}[]{\eq@provide@sqr}
6851 \eq@define@key{provide}{ang}[]{\eq@provide@ang}
6852 \eq@define@key{provide}{eqref}[]{\eq@provide@eqref{#1}}
6853 \eq@define@key{provide}{tagform}[]{\eq@provide@tagform}
6854 \eq@define@key{provide}{maketag}[]{\eq@provide@maketag}

```

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

```

6855 \newcommand{\eqnlinesprovide}[1]{%
6856 \eq@setkeys{provide}{#1}%
6857 \ignorespaces
6858 }

```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6859 \let\eql@error@packageoption\@gobble
```

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

```
6860 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
6861 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
6862 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
6863 \eql@defaults@eqnlines
6864 \eql@equations@columns@set
6865 \eql@box@columns@set
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the T_EX conditional parsing mechanism:

```
6866 \ifdefined\tagsleft@true\else
6867   \expandafter\newif\csname iftagsleft@\endcsname
6868 \fi
6869 \ifdefined\@fleqntrue\else
6870   \expandafter\newif\csname if@fleqn\endcsname
6871 \fi
```

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

```
6872 \eql@amsmath@after{%
6873   \ifnum\eql@provide@opt@env=\tw@
6874     \iftagsleft@
6875       \eqnlineset{tags=left}%
6876     \else
6877       \eqnlineset{tags=right}%
6878     \fi
6879   \if@fleqn
6880     \eqnlineset{layout=left}%
6881   \else
6882     \eqnlineset{layout=center}%
6883   \fi
6884 \fi
6885 }
```

Process package options:

```
6886 \ProcessOptions
```

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

```
6887 \def\eql@error@packageoption#1{%
6888   \eql@error{may only use '#1' as a package option}%
6889 }
```

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

```

6890 \ifnum\eql@provide@opt@env=\tw@
6891 \ifdefined\eql@provide@opt@matrix
6892   \let\eql@provide@opt@amsmathpatch\eql@false
6893 \fi\fi
6894 \ifdefined\eql@provide@opt@backup
6895   \let\eql@provide@opt@amsmathpatch\eql@true
6896 \fi
6897 \ifdefined\eql@provide@opt@amsmathpatch
6898   \eql@amsmath@fixends
6899   \eql@amsmath@fixmatrices
6900 \fi

```

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

```

6901 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi

```

Provide native \LaTeX environment `equation` and symbolic shortcut `\[...\]` if desired:

```

6902 \ifnum\eql@provide@opt@env>\z@
6903   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6904 \fi

```

Provide amsmath equation environments if desired:

```

6905 \ifnum\eql@provide@opt@env=\tw@
6906   \eqnlinesprovide{%
6907     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6908     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6909     multlined,gathered,aligned,alignedat,%
6910     subequations}
6911 \fi

```

Provide symbolic shortcut `\<...\>` if desired:

```

6912 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi

```

Provide equation reference `\eqref` if desired:

```

6913 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi

```

Provide `cases` and `matrix` environments if desired:

```

6914 \ifdefined\eql@provide@opt@matrix
6915   \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
6916 \fi

```