

The `eqnlines` Package

Source Code Documentation

Niklas Beisert

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

2026/01/14, v0.13

<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	3
2.2	Supporting Definitions	4
2.3	Dollardollar Abstraction	4
2.4	Look-Ahead in Alignment	4
2.5	Error Messages	7
2.6	<code>amsmath</code> Integration	8
2.7	PDF Tagging Support	8
2.8	Key-Value Processing	10
3	Parameters and Registers	12
3.1	Parameters	12
3.2	Registers	15
3.3	Hooks	17
4	Features	18
4.1	Punctuation	18
4.2	Math Classes at Alignment	20

4.3	Framed Cells	21
4.4	Single-Line Composition	21
4.5	Alternative Content Description	22
5	Equation Numbering	22
5.1	Supporting Definitions	22
5.2	Schemes	23
5.3	Interface	27
5.4	Integration	32
5.5	Positioning	34
5.6	Component Display	38
5.7	Tag Composition	41
6	Subequation Numbering	44
6.1	Definitions	44
6.2	Environment	45
6.3	Subequation Scheme	46
7	Display Equations Support	47
7.1	Display Breaks	47
7.2	Explicit Vertical Space	48
7.3	Default Vertical Spacing	50
7.4	Entry and Exit	51
7.5	Stack	56
8	Multi-Line Support	57
8.1	Measure Support	57
8.2	Line Breaks	59
8.3	Intertext	60
8.4	Line Marks	62
9	Column Placement	63
9.1	Supporting Definitions	63
9.2	Shape Schemes	64
9.3	Width Data	67
9.4	Best Line Selection	71
9.5	Tag Margin	72
9.6	Single Column	72
9.7	Multiple Columns	74
10	Single Column Arrangement	79
10.1	Supporting Definitions	79
10.2	Arrangement Methods	80
10.3	Central Alignment	81
10.4	Left Alignment	83
10.5	Right Alignment	84
11	Equations Box Environment	85
11.1	Line Breaks	85
11.2	Stacked Mode	86
11.3	Aligned Mode	87
11.4	Main	88
11.5	Environment	90
12	Single-Line Equation	91

12.1 Native Mode	91
12.2 Print	91
13 Multi-Line with Single Column	93
13.1 Measure	93
13.2 Column Placement	94
13.3 Print	95
14 Multi-Line with Multiple Columns	96
14.1 Support	96
14.2 Transpose	97
14.3 Measure	99
14.4 Columns Placement	101
14.5 Print	102
15 Interface	105
15.1 Scanning the Equation Body	105
15.2 Options Processing	108
15.3 Single-Line Main	110
15.4 Multi-Line Main	111
15.5 Equations Environment	113
16 Options	116
16.1 Selection Tools	116
16.2 Options Declarations	117
16.3 Parameter Presets	128
16.4 Component Selection	130
16.5 Global and Package Options	139

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\eq1@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 \@currenvir}
11 \def\eql@verbose@msg@leaveenv{ leaving \@currenvir}
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{starting line \the\eql@row@}

```

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:

```

18 \let\eql@false\@undefined
19 \let\eql@true\@empty

```

TODO: describe

```

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

```

2.3 Dollardollar Abstraction

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

```

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

`\eq@ifnextchar@loose` 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:

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

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

```
44 \let\eq@atxi=@
45 \begingroup
46   \makeatother
47   \let\tmp=@%
48   \makeatletter
49   \global\let\eq@atxii\tmp
50 \endgroup
```

`\eq@ifnextgobble@...` We introduce a collection of look-ahead macros which do or do not skip over spaces. The macros `\eq@ifstar@...` and `\eq@testopt@...` replicate the L^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 [...]:

```
51 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
52 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
53 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
54 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
55 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{#2}{%
56   \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
57 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{#2}{%
58   \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}}
59 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[{#1}{#1}]{#2}}
60 \long\def\eq@testopt@tight#1#2{\eq@ifnextchar@tight[{#1}{#1}]{#2}}
61 \long\def\eq@teststaropt@loose#1#2#3{%
62   \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}}
63 \long\def\eq@teststaropt@tight#1#2#3{%
64   \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}}
65 \long\def\eq@teststaroropt@loose#1#2#3{%
66   \eq@ifstar@loose{#1}{\eq@testopt@loose{#2}{#3}}}
67 \long\def\eq@teststaroropt@tight#1#2#3{%
```

```

68 \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}
69 \long\def\eq@gobbleopt[#1]{}
70 \long\def\eq@gobbleoptone[#1]#2{}

```

TODO: describe

```

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

```

TODO: describe

```

72 \def\eq@parseopt#1#2{%
73   \def\eq@parseopt@case{#1}%
74   \def\eq@parseopt@end{#2}%
75   \eq@parseopt@peek
76 }
77 \def\eq@parseopt@peek{%
78   \futurelet\eq@parseopt@token\eq@parseopt@select
79 }
80 \def\eq@parseopt@select{%
81   \let\eq@parseopt@next\eq@parseopt@other
82   \ifx\eq@parseopt@token@sptoken
83     \let\eq@parseopt@next\eq@parseopt@end
84   \fi
85   \eq@parseopt@case
86   \eq@parseopt@next
87 }
88 \def\eq@parseopt@other{\eq@parseopt@warn\eq@parseopt@end}
89 \let\eq@parseopt@warn\@empty
90 \def\eq@parseopt@gobble#1{\eq@parseopt@peek}

```

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

```

91 \def\eq@spbgroup{\iffalse{\fi\ifnum0='}\fi}
92 \def\eq@speggroup{\ifnum0='{ \fi\iffalse}\fi}
93 \def\eq@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
94 \def\eq@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}

```

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

```

95 \long\def\eq@ampprotect#1#2{\eq@srbgroup#1{\eq@sregroup#2}}
96 \long\def\eq@ampprotecttwo#1#2#3{%
97   \eq@srbgroup#1{\eq@sregroup#2}{\eq@sregroup#3}}

```

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

```

98 \def\eq@ifnextchar@loose@ampsafe#1{%
99   \eq@ampprotecttwo{\eq@ifnextchar@loose#1}}
100 \def\eq@ifnextchar@tight@ampsafe#1{%
101   \eq@ampprotecttwo{\eq@ifnextchar@tight#1}}
102 \def\eq@ifstar@loose@ampsafe{\eq@ampprotecttwo\eq@ifstar@loose}
103 \def\eq@ifstar@tight@ampsafe{\eq@ampprotecttwo\eq@ifstar@tight}

```

¹See <https://www.latex-project.org/cgi-bin/ltbugs2html?pr=latex/3040>,
<https://www.latex-project.org/cgi-bin/ltbugs2html?pr=amslatex/1834> and
<https://tex.stackexchange.com/questions/9897/showcase-of-brace-tricks-egroup-iffalse-fi-etc>.

```

104 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
105 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
106 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
107 \long\def\eql@teststaropt@tight@ampsafe{%
108   \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:

```

109 \let\eql@ifnextchar@org\@ifnextchar
110 \let\eql@ifstar@org\@ifstar
111 \let\eql@testopt@org\@testopt
112 \def\eql@amprevert{%
113   \let\@ifnextchar\eql@ifnextchar@org
114   \let\@testopt\eql@testopt@org
115   \let\@ifstar\eql@ifstar@org
116 }
117 \def\eql@amproof{%
118   \let\@ifnextchar\eql@ifnextchar@loose@ampsafe
119   \let\@testopt\eql@testopt@loose@ampsafe
120   \let\@ifstar\eql@ifstar@loose@ampsafe
121 }

```

2.5 Error Messages

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

```

122 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
123 \def\eql@warning{\PackageWarning{eqnlines}}

```

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

```

124 \def\eql@warn@here#1{\eql@warning{\string#1 not allowed outside equations}}
125 \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`

```

126 \def\eql@warn@tags@unused#1#2{\eql@warning{Unused equation #1:
127   #2 will be lost}}
128 \def\eql@warn@tags@multiple#1#2#3{\eql@warning{Multiple equation #1:
129   previous #2 will be lost#3}}
130 \def\eql@warn@label@unused{\eql@warn@tags@unused{\string\label}}
131   {label '\eql@tags@label'}}
132 \def\eql@warn@label@multiple#1{\eql@warn@tags@multiple{\string\label's}
133   {label '\eql@tags@label'}{ and replaced by '#1'}}
134 \def\eql@warn@name@unused{\eql@warn@tags@unused{label name}
135   {name declaration}}
136 \def\eql@warn@name@multiple{\eql@warn@tags@multiple{label names}
137   {name declaration}{}}
138 \def\eql@warn@tag@unused{\eql@warn@tags@unused{\string>tag}
139   {tag declaration}}
140 \def\eql@warn@tag@multiple{\eql@warn@tags@multiple{\string>tag's}
141   {tag declaration will be lost}{}}
142 \def\eql@warn@ref@unused{\eql@warn@tags@unused{tag label}
143   {tag label declaration}}
144 \def\eql@warn@ref@multiple{\eql@warn@tags@multiple{tag labels}

```

```

145     {tag label declaration}{}}

146 \def\eql@warn@parseopt{%
147   \eql@warning{Unknown modifier token: modifier parsing stopped}}
148 \def\eql@warn@parseopt@verbose{%
149   \eql@warning{Unknown modifier token: '\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/`:

```

150 \def\eql@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
151 \def\eql@amsmath@before#1{%
152   \ifpackageloaded{amsmath}{\AddToHook{package/amsmath/before}{#1}}
153 \def\eql@amsmath@undefine#1{\eql@amsmath@before{\let#1\undefined}}
154 \def\eql@amsmath@let#1#2{\eql@amsmath@undefine#1\let#1#2}

```

TODO: temporary fix for development stages

```

155 \ifpackageloaded{amsmath}{\{
156   \DeclareHookRule{package/amsmath/after}
157   {eqnlines}{after}{latex-lab-testphase-math}}

```

2.7 PDF Tagging Support

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

```

158 \let\eql@tagging@on\eql@false
159 \IfFormatAtLeastTF{2025-06-01}{%
160   \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}{}
161 \ifdefined\eql@tagging@on
162   \def\eql@tagging@mathsave{%
163     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
164   \def\eql@tagging@mathaddlast{%
165     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
166   \def\eql@tagging@tagbegin{%
167     \UseTaggingSocket{math/display/tag/begin}}
168   \def\eql@tagging@tagend{%
169     \UseTaggingSocket{math/display/tag/end}}
170   \def\eql@tagging@tagsave{%
171     \UseTaggingSocket{math/luamml/mtable/tag/save}}
172   \def\eql@tagging@tagaddbox{%
173     \setbox\z@\copy\eql@tagbox%
174     \UseTaggingSocket{math/luamml/mtable/tag/set}}
175   \def\eql@tagging@tablesaveinner{%
176     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
177   \def\eql@tagging@tableaddinner{%
178     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
179   \def\eql@tagging@tablesaveinner{%

```

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


```

180 \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
181 \def\eql@tagging@tablesalign{%
182 \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
183 \def\eql@tagging@alignleft{%
184 \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
185 \def\eql@tagging@aligncenter{%
186 \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
187 \def\eql@tagging@alignright{%
188 \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:

```

189 \let\eql@single@doscan\eql@true
190 \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:

```

191 \def\eql@tagging@start{%
192 \csname bool_if:N\expandafter\endcsname
193 \csname l__math_collected_bool\endcsname{%
194 \edef\eql@tmp{{\@currentvir}{\unexpanded\expandafter{\eql@tagging@opt}}}%
195 \the\eql@scan@reg@}%
196 \csname __math_process:nn\expandafter\endcsname\eql@tmp
197 \@kernel@math@registered@begin
198 \csname bool_set_true:N\expandafter\endcsname
199 \csname l__math_collected_bool\endcsname
200 }%
201 }
202 \def\eql@tagging@end{}
203 \def\eql@tagging@register@luamml#1{%
204 \AddToHook{package/luamml/after}{%
205 \eql@letcs{c__luamml_label_#1_tl}{\empty}}
206 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}

```

When tagging is deactivated, provide empty definitions:

```

207 \else
208 \let\eql@tagging@mathsave\@empty
209 \let\eql@tagging@mathaddlast\@empty
210 \let\eql@tagging@tagbegin\@empty
211 \let\eql@tagging@tagend\@empty
212 \let\eql@tagging@tagsave\@empty
213 \let\eql@tagging@tagaddbox\@empty
214 \let\eql@tagging@tablesaverinner\@empty
215 \let\eql@tagging@tableaddinner\@empty
216 \let\eql@tagging@tablesavelines\@empty
217 \let\eql@tagging@tablesalign\@empty
218 \let\eql@tagging@alignleft\@empty
219 \let\eql@tagging@aligncenter\@empty
220 \let\eql@tagging@alignright\@empty
221 \let\eql@tagging@start\@empty
222 \let\eql@tagging@end\@empty
223 \let\eql@tagging@register@luamml\@gobble

```

```

224 \let\eql@tagging@register@env\@gobble
225 \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:

```

226 \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:

```

227 \def\eql@decide@relax{\eql@tmpb:=\relax}
228 \def\eql@decide@select#1#2#3#4{%
229   \def\eql@tmpa{#3}%
230   \let\eql@tmpd\@undefined
231   \@for\eql@tmpc=#4\do{%
232     \ifdefined\eql@tmpd\else
233       \edef\eql@tmpb{\noexpand\eql@tmpb:=\expandafter\@firstoftwo\eql@tmpc}%
234       \ifx\eql@tmpb\eql@decide@relax
235         \def\eql@tmpa{\relax}%
236       \fi
237       \expandafter\@for\eql@tmpb\do{%
238         \ifx\eql@tmpa\eql@tmpb
239           \edef\eql@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
240             \expandafter\@secondoftwo\eql@tmpc}}%
241         \fi
242       }%
243     \fi
244   }%
245   \ifdefined\eql@tmpd
246     \eql@tmpd
247   \else
248     \eql@error{undefined value '3' for option '#2' of '#1'}%
249   \fi
250 }

```

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

```

251 \def\eql@decide@true{on,true,yes,enabled}
252 \def\eql@decide@false{off,false,no,disabled}

```

`\eql@decide@if`

```

253 \def\eql@decide@if#1#2#3#4#5{%
254   \eql@decide@select{#1}{#2}{#3}{#4}{#5}%

```

```

255     {\eqld@decide@true{#4}},%
256     {\eqld@decide@false{#5}}}}

```

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

```

257 \def\eqld@decide@bool#1#2#3#4{%
258   \eqld@decide@if{#1}{#2}{#3}{\let#4\eqld@true}{\let#4\eqld@false}}

```

Key Declaration.

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

```

259 \def\eqld@define@key#1#2{%
260   \eqld@ifnextchar@loose[%]
261     {\eqld@define@key@opt{#1}{#2}}%
262     {\eqld@define@key@noopt{#1}{#2}}%
263 }
264 \def\eqld@define@key@noopt#1#2#3{\eqld@define@key@for{#1}{#2}{#3}}
265 \def\eqld@define@key@opt#1#2[#3]#4{\eqld@define@key@for{#1}{#2}{[#3]{#4}}}
266 \def\eqld@define@key@for#1#2#3{%
267   \def\eqld@for@fn##1##2##3{\define@key{eqld@##3}{##2}{#3}}%
268   \edef\eqld@for@vara{\noexpand\eqld@for@vara:=#1}%
269   \expandafter\@for\eqld@for@vara\do{%
270     \edef\eqld@for@varb{\noexpand\eqld@for@varb:=#2}%
271     \expandafter\@for\eqld@for@varb\do{%
272       \edef\eqld@for@call##1{%
273         \noexpand\eqld@for@fn{##1}{\eqld@for@varb}{\eqld@for@vara}}%
274       \eqld@for@call{##1}%
275     }%
276   }%
277 }

```

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

```

278 \def\eqld@setkeys#1#2{%
279   \def\eqld@tmp{\setkeys{eqld@#1}}%
280   \expandafter\eqld@tmp\expandafter{#2}%
281 }

```

Options and Control Interface.

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

```

282 \let\eqld@nextopt\@empty
283 \def\eqld@nextopt@process#1{%
284   \eqld@setkeys{#1}\eqld@nextopt
285   \let\eqld@tagging@opt\eqld@nextopt
286   \global\let\eqld@nextopt\@empty
287 }

```

`\eqnaddopt`

```

288 \newcommand{\eqnaddopt}[1]{%

```

```

289 \ifx\eql@nextopt\@empty
290   \eql@append\eql@nextopt{#1}%
291 \else
292   \eql@append\eql@nextopt{, #1}%
293 \fi
294 }

```

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

```

295 \newcommand{\eqnlineset}[1]{%
296   \eql@setkeys{setup}{#1}%
297   \ignorespaces
298 }

```

`\eql@control@default`

```

299 \protected\def\eql@control@default{%
300   \eql@warn@here\eqncontrol
301   \@gobble
302 }
303 \let\eqncontrol\eql@control@default

```

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

```

304 \newcommand{\eql@control}[1]{%
305   \relax
306   \eql@setkeys{control}{#1}%
307   \ignorespaces
308 }

```

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:

```

309 \let\eql@tagsleft\eql@false

```

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

```

310 \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`:

```

311 \def\eql@layoutleftmargin{\leftmargini}
312 \def\eql@layoutleftmarginmax{.5\maxdimen}
313 \def\eql@layoutleftmarginmin{\z@}

```

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

```
314 \newdimen\eql@tagmargin@
315 \let\eql@tagmargin@val\@undefined
316 \newdimen\eql@tagmargin@ratio@
317 \eql@tagmargin@ratio@\p@
318 \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:

```
319 \newdimen\eql@indent@
320 \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:

```
321 \newdimen\eql@paddingleft@
322 \newdimen\eql@paddingright@
323 \let\eql@paddingleft@val\@undefined
324 \let\eql@paddingright@val\@undefined
```

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

```
\eql@display@marginleft
\eql@display@marginright
325 \let\eql@display@linewidth\@undefined
326 \let\eql@display@marginleft\@undefined
327 \let\eql@display@marginright\@undefined
```

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

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

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

```
\eql@break@line@shortsep
\eql@break@col@sep
\eql@break@col@shortsep
329 \def\eql@break@line@sep{2em minus 1em}
330 \def\eql@break@line@shortsep{1em}
331 \def\eql@break@col@sep{2em minus 1em}
332 \def\eql@break@col@shortsep{1em}
```

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

```
333 \def\eql@spread@val{\jot}
334 \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.

```
335 \newdimen\eql@tagfuzz@
336 \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:

```
337 \def\eqldisplay@height{\@undefined
338 \def\eqldisplay@depth{\@undefined
```

`\eqlskip@mode@short` The setting `\eqlskip@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:

```
339 \def\eqlskip@mode@short{2}

340 \def\eqlskip@mode@cont@above{2}
341 \def\eqlskip@mode@cont@below{0}

342 \def\eqlskip@mode@par@above{3}
343 \def\eqlskip@mode@par@below{0}

344 \def\eqlskip@mode@top@above{4}
345 \def\eqlskip@mode@top@below{0}

346 \newcount\eqlskip@mode@leave@
347 \let\eqlskip@force@leave\@undefined
```

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

`\eqlskip@force@below`
`\mode@above@ (counter)`
`\mode@below@ (counter)`

```
348 \newcount\eqlskip@mode@above@
349 \newcount\eqlskip@mode@below@
350 \let\eqlskip@force@above\@undefined
351 \let\eqlskip@force@below\@undefined
352 \let\eqlskip@custom@above\@undefined
353 \let\eqlskip@custom@below\@undefined
```

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

`\eqlskip@top@above` The glue when an equation is at the top of a vertical list is specified by

`\eqlskip@top@below` `\eqlskip@top@above` and `\eqlskip@top@below`:

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

`\eqlskip@med@above` The surrounding glue for an equation with reduced spacing is given by

`\eqlskip@med@below` `\eqlskip@med@above` and `\eqlskip@med@below`:

```
354 \def\eqlskip@long@above{\abovedisplayskip}
355 \def\eqlskip@long@below{\belowdisplayskip}
356 \def\eqlskip@short@above{\abovedisplayshortskip}
357 \def\eqlskip@short@below{\belowdisplayshortskip}
358 \def\eqlskip@cont@above{\eqlskip@short@above}
359 \def\eqlskip@cont@below{\eqlskip@short@below}
360 \def\eqlskip@par@above{\eqlskip@long@above}
361 \def\eqlskip@par@below{\eqlskip@long@below}
362 \def\eqlskip@top@above{\eqlskip@long@above}
363 \def\eqlskip@top@below{\eqlskip@long@below}
364 \def\eqlskip@med@above{\abovedisplayskip/2}
365 \def\eqlskip@med@below{\belowdisplayskip/2}
366 \def\eqlskip@tag@above{\z@skip}
367 \def\eqlskip@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:

```
368 \newdimen\eq@colsepmin@
369 \def\eq@colsepmin@val{1em}
370 \def\eq@colsepmax@val{.5\maxdimen}
```

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

```
371 \newdimen\eq@tagwidthmin@
372 \eq@tagwidthmin@{\z@}
```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. \TeX '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.

```
373 \newdimen\eq@tagsepmin@
374 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}
```

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

```
\eq@equations@ang@opt
\eq@box@ang@opt
375 \def\eq@equations@sqr@opt{equation,nonumber}
376 \def\eq@equations@ang@opt{align,nonumber}
377 \def\eq@box@ang@opt{align}
```

Multi-Line Align Mode.

```
378 \let\eq@columns@fulllength\eq@false
```

3.2 Registers

TODO: describe

General. **TODO:** describe

```
379 \newcount\eq@count@
380 \newdimen\eq@dimen@
381 \newskip\eq@skip@
```

TODO: describe

```
382 \let\eq@display@container\@empty
```

`\eq@cellbox@` (*box*) The box `\eq@cellbox@` holds the present alignment component and `\eq@tagbox@` the tag for the present line. The corresponding dimensions `\eq@cellwidth@` and

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

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

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

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

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

```
383 \newbox\eq@cellbox@
384 \newbox\eq@tagbox@
385 \newdimen\eq@cellwidth@
386 \newdimen\eq@prevwidth@
387 \newdimen\eq@tagwidth@
```

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

```

388 \newdimen\eql@prevdepth@
389 \newcount\eql@prevgraf@

```

```

\eql@totalwidth@ (dimen)
\eql@tagwidth@max@ (dimen)
\eql@totalheight@ (dimen)
390 \newdimen\eql@totalwidth@
391 \newdimen\eql@tagwidth@max@
392 \newdimen\eql@totalheight@
393 \newdimen\eql@topheight@
394 \newdimen\eql@bottomdepth@

```

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

```

395 \newdimen\eql@line@height@
396 \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)
397 \newdimen\eql@line@width@
398 \newdimen\eql@line@avail@
399 \newdimen\eql@line@pos@
400 \newcount\eql@line@availsep@
401 \newcount\eql@line@widthsep@
402 \newcount\eql@line@possep@
403 \newdimen\eql@line@offset@
404 \newdimen\eql@line@prevdepth@
405 \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)
406 \newcount\eql@row@
407 \newcount\eql@totalrows@
408 \newcount\eql@tagrows@

```

```

\eql@column@
\eql@totalcolumns@
409 \newcount\eql@column@
410 \newcount\eql@totalcolumns@

```

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

```

411 \newdimen\eql@colsep@

```

```

\eql@intercolumns@ (counter)
412 \newcount\eql@intercolumns@

```

Vertical Spacing Adjustments.

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

```
413 \newdimen\eqldisplay@firstavail@
414 \def\eqldisplay@firstavail@set#1{%
415   \ifnum\eqldisplay@row@=\@ne
416     \global\eqldisplay@appendexpand\eqldisplay@container{%
417       \eqldisplay@firstavail@the#1\relax}%
418   \fi
419 }
```

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

```
420 \newdimen\eqldisplay@aboveextend@
421 \newdimen\eqldisplay@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:

```
422 \ifdefined\measuring@true\else
423   \expandafter\newif\csname ifmeasuring@\endcsname
424 \fi
425 \AddToHook{package/hyperref/after}{
426   \ifdefined\Hy@ifnotmeasuring
427     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
428   \fi
429 }
```

`\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:

```
430 \ifdefined\@displaytrue\else
431   \expandafter\newif\csname if@display\endcsname
432   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
433 \fi
```

3.3 Hooks

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

```
434 \let\eqldhook@blockbefore\@empty
435 \let\eqldhook@blockafter\@empty
436 \let\eqldhook@blockin\@empty
437 \let\eqldhook@blockout\@empty
```

```

438 \let\eqL@hook@linein\@empty
439 \let\eqL@hook@lineout\@empty
440 \let\eqL@hook@colin\@empty
441 \let\eqL@hook@colout\@empty
442 \let\eqL@hook@eqin\@empty
443 \let\eqL@hook@eqout\@empty
444 \let\eqL@hook@innerleft\@empty
445 \let\eqL@hook@innerright\@empty
446 \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. A value
`\eqL@punct@line` `\relax` indicates that the punctuation should be handed down to the next lower level:
`\eqL@punct@block` **TODO:** update

```

447 \let\eqL@punct@col\@empty
448 \let\eqL@punct@line\relax
449 \let\eqL@punct@block\relax
450 \let\eqL@punct@main\relax

```

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

```

451 \let\eqL@punct@sep\relax

```

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

```

452 \def\eqL@punct@relax{\relax}
453 \def\eqL@punct@tilde{~}
454 \def\eqL@punct@set#1#2{%
455   \def#1{#2}%
456   \ifx#1\eqL@punct@relax
457     \let#1\relax
458   \fi
459   \ifx#1\eqL@punct@tilde
460     \let#1\@empty
461   \fi
462 }

```

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

```

463 \def\eqnpunct{%
464   \eqL@ifstar@tight\eqL@punct@next@setrelax\eqL@punct@next@set}
465 \def\eqL@punct@next@set#1{%
466   \ifmmode
467     \eqL@punct@set\eqL@punct@col{#1}%
468     \eqL@punct@set\eqL@punct@line{#1}%
469     \eqL@punct@set\eqL@punct@block{#1}%
470   \else

```

```

471 \eqnaddopt{punct={#1}}%
472 \fi
473 \ignorespaces}
474 \def\eql@punct@next@setrelax{%
475 \ifmmode
476 \let\eql@punct@block\relax
477 \else
478 \eqnaddopt{punct*}%
479 \fi
480 \ignorespaces}

```

`\eql@punct@print@col` Output the punctuation for the present column. If non-empty, prepend some separation:

```

481 \def\eql@punct@print@col{%
482 \ifx\eql@punct@col\@empty\else
483 \eql@punct@sep
484 \eql@punct@col
485 \fi
486 }

```

`\eql@punct@apply@col` Output the punctuation for the present column. Clear the punctuation so that no further column punctuation is output within the current group:

```

487 \def\eql@punct@apply@col{%
488 \eql@punct@print@col
489 \let\eql@punct@col\@empty
490 }

```

Output the punctuation currently set for lines unless disabled:

`\eql@punct@print@line`

```

491 \def\eql@punct@print@line{%
492 \ifx\eql@punct@line\relax
493 \eql@punct@print@col
494 \else
495 \ifx\eql@punct@line\@empty\else
496 \eql@punct@sep
497 \eql@punct@line
498 \fi
499 \fi
500 }

```

Output the punctuation currently set for lines unless disabled. Alike `\eql@punct@apply@col` prevent further output of punctuations for lines and columns within the current group:

`\eql@punct@apply@line`

```

501 \def\eql@punct@apply@line{%
502 \ifx\eql@punct@line\relax
503 % \TODO hand down immediately?
504 \else
505 \ifx\eql@punct@line\@empty\else
506 \eql@punct@sep
507 \eql@punct@line
508 \fi
509 \let\eql@punct@line\relax
510 \let\eql@punct@col\@empty
511 \fi

```

512 }

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

```
513 \def\eql@punct@apply@block{%
514   \ifx\eql@punct@block\relax
515   % \TODO hand down immediately?
516   \else
517     \ifx\eql@punct@block\@empty\else
518       \eql@punct@sep
519       \eql@punct@block
520     \fi
521     \let\eql@punct@block\relax
522     \let\eql@punct@line\relax
523     \let\eql@punct@col\@empty
524   \fi
525 }

526 \let\eqnpunctapply\eql@punct@apply@block
```

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:

```
527 \def\eql@class@innerright@sel{%
528   \ifdim\eql@prevwidth@=\z@
529     \eql@class@innerlead
530   \else
531     \eql@class@innerright
532   \fi
533 }
```

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

```
534 \def\eql@class@innerleft@set#1{%
535   \def\eql@class@innerleft{#1}%
536 }
537 \def\eql@class@innerright@set#1{%
538   \def\eql@class@innerright{#1}%
539   \let\eql@class@innerright@sel\eql@class@innerright
540 }
541 \def\eql@class@innerlead@set#1{%
542   \def\eql@class@innerlead{#1}%
543   \let\eql@class@innerright@sel\eql@class@innerright@sel@
544 }
```

`\eql@class@ampeq` We define two standard combinations of math classes intended to be used with ‘&=’
`\eql@class@eqamp` (ampeq) or ‘&’ (eqamp). The default setting is ‘&=’ (ampeq):

```
545 \def\eql@class@ampeq{%
546   \eql@class@innerleft@set{}%
547   \eql@class@innerright@set{}}%
548 }
```

```

549 \def\eql@class@eqamp{%
550   \eql@class@innerleft@set{\mathrel{}}}%
551   \eql@class@innerright@set{\mathrel{}}}%
552   \eql@class@innerlead@set{}}}%
553 }
554 \eql@class@ampeq

```

4.3 Framed Cells

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

```

555 \let\eql@frame@cmd@\undefined
556 \newdimen\eql@frame@margin@
557 \def\eql@frame@set[#1]{%
558   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
559 \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbbox}
560 \def\eql@frame@measure{%
561   \setbox\z@\hbox{\eql@frame@cmd{}}}%
562   \eql@frame@margin@.5\wd\z@
563 }
564 \def\eql@frame@print{%
565   \setbox\eql@cellbox@\hbox{%
566     \eql@frame@cmd{\unhbox\eql@cellbox@}%
567   }%
568 }
569 \def\eql@frame@adjust{%
570   \setbox\eql@cellbox@\hbox{%
571     \eql@frame@cmd{%
572       \unhbox\eql@cellbox@
573       \unkern
574       \unskip
575     }%
576     \hfil
577     \kern\z@
578   }%
579 }

```

4.4 Single-Line Composition

TODO: describe

`\eql@break@line`

```

580 \def\eql@break@line{\eql@srbgroup
581   \eql@ifnextgobble@tight~\eql@break@line@star
582   {\eql@punct@print@line\eql@break@line@star}}
583 \def\eql@break@line@star{%
584   \eql@ifstar@tight
585   {\eql@break@opt[\eql@break@line@shortsep]}%
586   {\eql@testopt@tight\eql@break@opt\eql@break@line@sep}}

```

`\eql@break@col`

```

587 \def\eql@break@col{\eql@srbgroup
588   \eql@ifnextgobble@tight~\eql@break@col@star
589   {\eql@punct@print@col\eql@break@col@star}}
590 \def\eql@break@col@star{%

```

```

591 \eq@ifstar@tight
592   {\eq@break@opt[\eq@break@col@shortsep]}%
593   {\eq@testopt@tight\eq@break@opt\eq@break@col@sep}}

\eq@break@opt

594 \def\eq@break@opt[#1]{\eq@sregroup\hspace{\glueexpr#1\relax}}

\eq@break@join

595 \def\eq@break@join{\eq@srbgroup
596   \eq@ifstar@tight
597   {\eq@break@join@opt[\eq@break@col@shortsep]}%
598   {\eq@testopt@tight\eq@break@join@opt\eq@break@col@sep}}
599 \def\eq@break@join@opt[#1]#2{\eq@sregroup%
600   \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

\eqnsep TODO: expand to lines and columns mode
\eqnbreak
\eqnjoin 601 \def\eqnsep{\eq@break@col}
602 \def\eqnbreak{\eq@break@line}
603 \def\eqnjoin{\eq@break@join}

```

4.5 Alternative Content Description

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

```
604 \DeclareRobustCommand{\eqnalt}[2][{}]{}
```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```

605 \let\eq@tags@autolabel\eq@false
606 \let\eq@tags@autotag\eq@true
607 \let\eq@tags@warn\eq@true

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

609 \let\eq@tagpos@doconvert\eq@false

610 \def\eq@tagpos@snap{4pt}

```

Registers.

```

611 \let\eq@numbering@mode\@undefined

612 \let\eq@numbering@active\eq@true
613 \let\eq@numbering@multi\eq@true

```

```

614 \let\eql@tags@container\@undefined
615 \def\eql@tags@container@clear{%
616   \let\eql@tags@label\@undefined
617   \let\eql@tags@name\@undefined
618   \let\eql@tags@tag\@undefined
619   \let\eql@tags@ref\@undefined
620   \let\eql@tags@anchor\@empty
621   \eql@tagpos@shift@z@
622   \eql@tagpos@smashup@z@
623   \eql@tagpos@smashdown@z@
624   \let\eql@tagpos@reserve\eql@true
625 }

626 \let\eql@tags@label\@undefined
627 \let\eql@tags@name\@undefined
628 \let\eql@tags@tag\@undefined
629 \let\eql@tags@ref\@undefined
630 \let\eql@tags@frame@cmd\@firstofone

```

tags@glabel@ (*counter*)

```

631 \newcount\eql@tags@glabel@
632 \eql@tags@glabel@z@
633 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
634 \def\eql@tags@glabel@step{\global\advance\eql@tags@glabel@\@ne}

635 \let\eql@tagpos@continuous\eql@false

636 \newcount\eql@tagpos@row@
637 \newcount\eql@tagpos@prevrow@
638 \newdimen\eql@tagpos@shift@
639 \newdimen\eql@tagpos@smashup@
640 \newdimen\eql@tagpos@smashdown@
641 \newdimen\eql@tagpos@current@
642 \newdimen\eql@tagpos@plain@
643 \newdimen\eql@tagpos@raised@
644 \newdimen\eql@tagpos@target@
645 \newdimen\eql@tagpos@headroom@
646 \newdimen\eql@tagpos@footroom@

```

5.2 Schemes

TODO: describe

Table.

```

647 \def\eql@numbering@tab@sub{sub}
648 \def\eql@numbering@tab@all{all}
649 \def\eql@numbering@tab@first{first}
650 \def\eql@numbering@tab@last{last}
651 \def\eql@numbering@tab@in{in}
652 \def\eql@numbering@tab@out{out}
653 \def\eql@numbering@tab@middle{middle}
654 \def\eql@numbering@tab@best{best}
655 \def\eql@numbering@tab@here{here}
656 \def\eql@numbering@tab@top{top}
657 \def\eql@numbering@tab@bottom{bottom}

```

```

658 \def\eql@numbering@tab@center{center}
659 \def\eql@numbering@tab@centerone{centerone}
660 \def\eql@numbering@tab@median{median}
661 \def\eql@numbering@tab@baseline{baseline}

662 \let\eql@numbering@mode\eql@numbering@tab@all
663 \let\eql@numbering@mode@multi\eql@numbering@tab@all
664 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```

665 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
666 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
667 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
668 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
669 \let\eql@numbering@tab@outside\eql@numbering@tab@out
670 \let\eql@numbering@tab@inside\eql@numbering@tab@in
671 \let\eql@numbering@tab@within\eql@numbering@tab@in
672 \let\eql@numbering@tab@opt\eql@numbering@tab@best
673 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
674 \let\eql@numbering@tab@pick\eql@numbering@tab@here
675 \let\eql@numbering@tab@med\eql@numbering@tab@median
676 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
677 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone

```

TODO: describe

```

678 \let\eql@numbering@tab@a\eql@numbering@tab@all
679 \let\eql@numbering@tab@s\eql@numbering@tab@sub
680 \let\eql@numbering@tab@f\eql@numbering@tab@first
681 \let\eql@numbering@tab@l\eql@numbering@tab@last
682 \let\eql@numbering@tab@o\eql@numbering@tab@out
683 \let\eql@numbering@tab@i\eql@numbering@tab@in
684 \let\eql@numbering@tab@h\eql@numbering@tab@here
685 \let\eql@numbering@tab@t\eql@numbering@tab@top
686 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
687 \let\eql@numbering@tab@c\eql@numbering@tab@center
688 \let\eql@numbering@tab@m\eql@numbering@tab@median
689 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
690 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
691 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
692 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

693 \def\eql@numbering@init@all{\let\eql@numbering@mode\eql@numbering@tab@all}

```

TODO: describe

```

694 \def\eql@numbering@init@sub{%
695   \let\eql@numbering@mode\eql@numbering@tab@all
696   \ifdefined\eql@subequations@active
697     \let\eql@numbering@mode\eql@numbering@tab@all
698   \else
699     \let\eql@numbering@mode\eql@numbering@tab@sub
700   \fi
701 }

702 \def\eql@numbering@init@first{\eql@tagpos@row@{\@ne}}
703 \def\eql@numbering@init@last{\eql@tagpos@row@{\@MM}}
704 \def\eql@numbering@init@here{\eql@tagpos@row@{\m@ne}}

```


TODO: describe

```
705 \def\eq@numbering@init@in{%
706   \ifdefined\eq@tagsleft
707     \eq@numbering@init@last
708   \else
709     \eq@numbering@init@first
710   \fi
711 }
```

TODO: describe

```
712 \def\eq@numbering@init@out{%
713   \ifdefined\eq@tagsleft
714     \eq@numbering@init@first
715   \else
716     \eq@numbering@init@last
717   \fi
718 }
```

TODO: describe

```
719 \def\eq@tagpos@eval@middle{%
720   \ifnum\eq@tagpos@row@=\z@
721     \eq@tagpos@row@=\numexpr(\eq@totalrows@
722       +\ifdefined\eq@tagsleft\z@\else\@ne\fi)/\tw@\relax
723   \fi
724 }
```

TODO: describe

```
725 \def\eq@tagpos@eval@best{%
726   \ifnum\eq@tagpos@row@=\z@
727     \let\eq@numbering@best@use\eq@true
728     \eq@numbering@init@out
729   \fi
730 }
```

TODO: describe

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

TODO: describe

```
732 \let\eq@numbering@init@top\eq@numbering@init@continuous
733 \def\eq@tagpos@eval@top{%
734   \eq@tagpos@current@\z@
735 }
```

TODO: describe

```
736 \let\eq@numbering@init@bottom\eq@numbering@init@continuous
737 \def\eq@tagpos@eval@bottom{%
738   \eq@tagpos@current@\dimexpr\eq@totalheight@
739     -\eq@tagheight@block@-\eq@tagdepth@block@\relax
740 }
```

TODO: describe

```
741 \let\eq@numbering@init@center\eq@numbering@init@continuous
742 \def\eq@tagpos@eval@center{%
743   \ifnum\eq@totalrows@=\@ne
744     \eq@tagpos@row@\@ne
```

```

745 \fi
746 \eq\tagpos@current@\dimexpr(\eq\totalheight@
747   -\eq\tagheight@block@-\eq\tagdepth@block@)/\tw@\relax
748 }

```

TODO: describe

```

749 \let\eq\numbering@init@centerone\eq\numbering@init@continuous
750 \def\eq\tagpos@eval@centerone{%
751   \eq\tagpos@current@\dimexpr(\eq\totalheight@
752     -\eq\tagheight@block@-\eq\tagdepth@block@)/\tw@\relax
753 }

```

TODO: describe

```

754 \let\eq\numbering@init@baseline\eq\numbering@init@continuous
755 \def\eq\tagpos@eval@baseline{%
756   \eq\tagpos@current@\dimexpr(\eq\totalheight@
757     +\eq\topheight@-\eq\bottomdepth@)/\tw@-\eq\tagheight@block@\relax
758 }

```

TODO: describe

```

759 \let\eq\numbering@init@median\eq\numbering@init@continuous
760 \def\eq\tagpos@eval@median{%
761   \ifnum\eq\tagpos@row@=\z@
762     \ifodd\eq\totalrows@
763       \eq\tagpos@row@\numexpr(\eq\totalrows@+\@ne)/\tw@\relax
764     \else
765       \eq\tagpos@row@\numexpr(\eq\totalrows@+\tw@)/\tw@\relax
766       \eq\dimensions@get\eq\tagpos@row@
767       \advance\eq\tagpos@shift@\dimexpr\eq\line@height@
768         +(\eq\line@interline@-\eq\tagheight@block@
769           +\eq\tagdepth@block@)/\tw@\relax
770   \fi
771   \ifnum\eq\totalrows@=\@ne
772     \eq\tagpos@row@\@ne
773   \else
774     \eq\tagpos@adjust@eval@convert
775     \eq\tagpos@row@\z@
776   \fi
777 \fi
778 }

```

Selection.

```

779 \def\eq\numbering@set#1{%
780   \ifcsname eq\numbering@tab@#1\endcsname
781     \expandafter\let\expandafter\eq\numbering@mode
782     \csname eq\numbering@tab@#1\endcsname
783     \ifx\eq\numbering@mode\eq\numbering@tab@all
784       \let\eq\numbering@mode@multi\eq\numbering@mode
785     \else\ifx\eq\numbering@mode\eq\numbering@tab@sub
786       \let\eq\numbering@mode@multi\eq\numbering@mode
787     \else
788       \let\eq\numbering@mode@single\eq\numbering@mode
789     \fi\fi
790   \else
791     \eq\error{numbering mode '#1' unknown: setting mode to 'all'}%
792     \let\eq\numbering@mode\eq\numbering@tab@all

```

```

793 \fi
794 }

```

TODO: describe

```

795 \def\eql@numbering@init{%
796   \let\eql@numbering@multi\eql@false
797   \let\eql@tagpos@continuous\eql@false
798   \let\eql@numbering@subeq@use\eql@false
799   \let\eql@numbering@best@use\eql@false
800   \eql@tagpos@row@\z@
801   \csname eql@numbering@init@\eql@numbering@mode\endcsname
802   \ifdefined\eql@numbering@active
803     \let\eql@numbering@eqnswinit\@eqnswtrue
804   \else
805     \let\eql@numbering@eqnswinit\@eqnswfalse
806   \fi
807   \let\eql@numbering@active\eql@false
808 }

```

5.3 Interface

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

```

809 \eql@amsmath@after{
810   \let\eql@print@eqnum@default\print@eqnum
811   \let\eql@incr@eqnum@default\incr@eqnum
812 }

```

TODO: describe

```

813 \protected\def\donumber{%
814   \if@eqnsw\else
815     \global\@eqnswtrue
816     \ifx\print@eqn\@empty
817       \global\let\print@eqn\eql@print@eqnum@default
818     \fi
819     \ifx\incr@eqn\@empty
820       \global\let\incr@eqn\eql@incr@eqnum@default
821     \fi
822   \fi
823 }

```

TODO: reconsider operation

`\numberhere`

```

824 \protected\def\eql@numberhere{%
825   \ifdefined\eql@numbering@multi
826     \global\@eqnswtrue
827   \else
828     \global\eql@tagpos@row@\eql@row@
829   \fi
830 }

```

TODO: describe

`\numbernext`

```

831 \protected\def\eql@numbernext{%

```

```

832 \ifdefined\eql@numbering@multi
833   \global\@eqnswfalse
834 \else
835   \ifdefined\eql@tagpos@continuous\else
836     \ifnum\eql@tagpos@row@=\eql@row@
837       \global\advance\eql@tagpos@row@\@ne
838     \fi
839   \fi
840 \fi
841 }

```

Activation Trigger.

```

842 \def\eql@tags@autoenable{%
843   \global\@eqnswtrue
844   \ifnum\eql@tagpos@row@=\m@ne
845     \numberhere
846   \fi
847 }

```

Labels. **TODO:** describe

\eql@label@org

```

848 \let\eql@label@org\label

```

TODO: describe

```

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

```

TODO: describe

```

850 \protected\def\eql@label{%
851   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
852 }

```

TODO: describe

```

853 \def\eql@tags@add@labelname[#1]#2{%
854   \def\eql@tmp{#1}%
855   \ifx\eql@tmp\eql@testopt@default\else
856     \eql@tags@add@name{#1}%
857   \fi
858   \eql@tags@add@label{#2}%
859 }

```

TODO: describe

```

860 \def\eql@tags@set@label#1{%
861   \ifdefined\eql@tags@warn
862     \ifdefined\eql@tags@label
863       \eql@warn@label@multiple{#1}%
864     \fi
865   \fi
866   \def\eql@tags@label{#1}%
867 }

```

TODO: describe

```

868 \def\eql@tags@set@name#1{%

```

```

869 \ifdefined\eql@tags@warn
870   \ifdefined\eql@tags@name
871     \eql@warn@name@multiple
872   \fi
873 \fi
874 \def\eql@tags@name{#1}%
875 }

```

TODO: describe

```

876 \def\eql@tags@add@label#1{%
877   \ifdefined\eql@tags@autolabel
878     \eql@tags@autoenable
879   \fi
880   \global\eql@appendexpand\eql@tags@container{%
881     \noexpand\eql@tags@set@label{#1}}%
882 }

```

TODO: describe

```

883 \def\eql@tags@add@name#1{%
884   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
885   \global\eql@appendmacro\eql@tags@container\eql@tmp
886 }

```

TODO: describe

```

887 \def\eql@tags@addblock@label#1{%
888   \eql@appendexpand\eql@tags@container@block{%
889     \noexpand\eql@tags@set@label{#1}}%
890 }

```

TODO: describe

```

891 \def\eql@tags@addblock@name#1{%
892   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
893   \eql@appendmacro\eql@tags@container@block\eql@tmp
894 }

```

Tags. **TODO:** describe

`\eql@tag@default`

```

895 \protected\def\eql@tag@default{%
896   \eql@warn@here\tag
897   \eql@tag@gobble
898 }
899 \let\tag\eql@tag@default

```

`\eql@tag@gobble`

```

900 \def\eql@tag@gobble{%
901   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}

```

TODO: describe

```

902 \protected\def\eql@tag{%
903   \eql@ampprotecttwo\eql@teststaropt@tight
904   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
905   \eql@testopt@default
906 }

```

\eql@tags@add@tagref

```
907 \def\eql@tags@add@tagref[#1]#2{%
908   \def\eql@tmp{#1}%
909   \ifx\eql@tmp\eql@testopt@default\else
910     \eql@tags@add@ref{#1}%
911   \fi
912   \eql@tags@add@tag{#2}%
913 }
```

TODO: describe

```
914 \def\eql@tags@set@tag#1{%
915   \ifdefined\eql@tags@warn
916     \ifdefined\eql@tags@tag
917       \eql@warn@tag@multiple
918     \fi
919   \fi
920   \def\eql@tags@tag{#1}%
921 }
```

TODO: describe

```
922 \def\eql@tags@set@ref#1{%
923   \ifdefined\eql@tags@warn
924     \ifdefined\eql@tags@ref
925       \eql@warn@ref@multiple
926     \fi
927   \fi
928   \def\eql@tags@ref{#1}%
929 }
```

TODO: describe

```
930 \def\eql@tags@add@tag#1{%
931   \ifdefined\eql@tags@autotag
932     \eql@tags@autoenable
933   \fi
934   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
935   \global\eql@appendmacro\eql@tags@container\eql@tmp
936 }
```

TODO: describe

```
937 \def\eql@tags@add@ref#1{%
938   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
939   \global\eql@appendmacro\eql@tags@container\eql@tmp
940 }
```

\eql@tags@add@tagform@off

```
941 \def\eql@tags@add@tagform@off{%
942   \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
943 }
```

TODO: describe

```
944 \def\eql@tags@addblock@tag#1{%
945   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
946   \eql@appendmacro\eql@tags@container@block\eql@tmp
947 }
```

TODO: describe

```
948 \def\eql@tags@addblock@ref#1{%
949   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
950   \eql@appendmacro\eql@tags@container@block\eql@tmp
951 }
```

TODO: describe

```
952 \def\eql@tags@addblock@tagform@off{%
953   \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
954 }
```

Raise Tags.

\raisetag

```
955 \def\eql@raisetag@default{%
956   \eql@warn@here\raisetag
957   \eql@raisetag@gobble
958 }

959 \def\eql@raisetag@gobble{%
960   \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble
961 }
```

TODO: describe

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

963 \def\eql@raisetag{%
964   \eql@ampprotecttwo\eql@ifstar@tight\eql@tags@add@raiseshift\eql@raisetag@test
965 }

966 \def\eql@raisetag@test#1{%
967   \def\eql@tmpa{#1}\def\eql@tmpb{!}%
968   \ifx\eql@tmpa\eql@tmpb
969     \eql@tags@add@forceraise
970   \else
971     \eql@tags@add@raisesmash{#1}%
972   \fi
973 }

974 \def\eql@tags@add@raiseshift#1{%
975   \global\eql@appendexpand\eql@tags@container{%
976     \advance\eql@tagpos@shift@the\glueexpr#1\relax\relax}%
977 }

978 \def\eql@tags@add@raisesmash#1{%
979   \dimen@0\glueexpr#1\relax
980   \ifdim\dimen@<\z@
981     \global\eql@appendexpand\eql@tags@container{%
982       \advance\eql@tagpos@smashdown@the\dimen@\relax}%
983   \else
984     \global\eql@appendexpand\eql@tags@container{%
985       \advance\eql@tagpos@smashup@the\dimen@\relax}%
986   \fi
987 }

988 \def\eql@tags@add@forceraise{%
989   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
990 }
```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```
991 \def\eql@numbering@settools{%
992   \let\label\eql@label
993   \let\tag\eql@tag
994   \let\raisetag\eql@raisetag
995   \let\numberhere\eql@numberhere
996   \let\numbernext\eql@numbernext
997 }
```

TODO: not necessary anymore

```
998 \def\eql@numbering@settools@gobble{%
999   \let\label\eql@label@gobble
1000   \let\tag\eql@tag@gobble
1001   \let\raisetag\eql@raisetag@gobble
1002   \let\numberhere\relax
1003   \let\numbernext\relax
1004 }
```

```
1005 \def\eql@numbering@autoblock{%
1006   \begingroup
1007     \let\eql@tags@warn\eql@false
1008     \eql@tags@container@block
1009     \ifdefined\eql@tags@autolabel
1010       \ifdefined\eql@tags@label
1011         \global\@eqnswtrue
1012       \fi
1013     \fi
1014     \ifdefined\eql@tags@autotag
1015       \ifdefined\eql@tags@tag
1016         \global\@eqnswtrue
1017       \fi
1018     \fi
1019   \endgroup
1020 }
```

```
1021 \def\eql@numbering@warnunused{%
1022   \ifdefined\eql@tags@label
1023     \eql@warn@label@unused
1024   \fi
1025   \ifdefined\eql@tags@name
1026     \eql@warn@name@unused
1027   \fi
1028   \ifdefined\eql@tags@tag
1029     \eql@warn@tag@unused
1030   \fi
1031   \ifdefined\eql@tags@erf
1032     \eql@warn@ref@unused
1033   \fi
1034 }
```

Single Line. **TODO:** describe


```

1035 \def\eql@numbering@single@init{%
1036   \let\eql@numbering@multi\eql@false
1037   \eql@numbering@settools
1038   \eql@numbering@eqnswinit
1039   \eql@numbering@autoblock
1040   \global\let\eql@tags@container\eql@tags@container@block
1041   \let\eql@tags@warn\eql@true
1042 }

1043 \def\eql@numbering@single@eval{%
1044   \ifnum\eql@tagpos@row@=\m@ne
1045     \@eqnswfalse
1046   \fi
1047 }

```

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

```

1048 \def\eql@numbering@measure@init{%
1049   \eql@numbering@settools
1050   \ifdefined\eql@numbering@multi\else
1051     \eql@numbering@eqnswinit
1052     \eql@numbering@autoblock
1053   \fi
1054   \global\let\eql@tags@container\eql@tags@container@block
1055   \let\eql@tags@warn\eql@true
1056 }

```

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

```

1057 \def\eql@numbering@measure@line@begin{%
1058   \ifdefined\eql@numbering@multi
1059     \global\eql@numbering@eqnswinit
1060   \fi
1061 }

```

TODO: describe

```

1062 \def\eql@numbering@measure@blocktag{%
1063   \ifdefined\eql@numbering@multi
1064     \@eqnswfalse
1065   \else
1066     \ifnum\eql@tagpos@row@=\m@ne
1067       \@eqnswfalse
1068     \fi
1069     \ifnum\eql@totalrows@=\z@
1070       \@eqnswfalse
1071     \fi
1072   \fi
1073 }

```

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!

```

1074 \def\eql@numbering@print@init{%
1075   \let\eql@tags@warn\eql@false
1076   \ifdefined\eql@numbering@multi
1077     \eql@numbering@settools

```

```

1078   \global\let\eq\eq\tags@container\eq\tags@container@block
1079   \else
1080     \let\eq\eq\tags@container@block\eq\tags@container
1081     \eq\@numbering@settools@gobble
1082   \fi
1083 }

```

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

```

1084 \def\eq\@numbering@print@block@begin{%
1085   \ifdefined\eq\@numbering@multi\else
1086     \ifnum\eq\@tagpos@row@>\z@
1087       \eq\@tags@makeblockanchor
1088       \global\eq\@appendexpand\eq\@tags@container@block{%
1089         \def\noexpand\eq\@tags@anchor{%
1090           \unexpanded\expandafter{\eq\@tags@anchor}}}%
1091     \fi
1092   \fi
1093   \ifdefined\eq\@numbering@subeq@use
1094     \eq\@tags@printsubeqlabel
1095   \fi
1096 }

```

TODO: describe

```

1097 \def\eq\@numbering@print@line@begin{%
1098   \ifdefined\eq\@numbering@multi
1099     \global\eq\@numbering@eqnswinit
1100   \fi
1101 }

```

TODO: describe

```

1102 \def\eq\@numbering@print@line@eval{%
1103   \ifdefined\eq\@numbering@multi
1104     \if@eqnsw
1105       \eq\@tags@container
1106     \fi
1107   \else
1108     \ifnum\eq\@tagpos@row@=\eq\@row@
1109       \@eqnswtrue
1110       \eq\@tags@container@block
1111     \else
1112       \@eqnswfalse
1113     \fi
1114   \fi
1115 }

```

5.5 Positioning

TODO: describe

```

1116 \def\eq\@tagpos@single@eval{%
1117   \if@eqnsw
1118     \csname eq\@tagpos@eval@\eq\@numbering@mode\endcsname
1119     \ifnum\eq\@tagpos@row@>\@one
1120       \eq\@tagpos@row@\@one
1121     \fi
1122     \ifdefined\eq\@tagpos@doconvert
1123       \let\eq\@tagpos@continuous\eq\@true

```

```

1124 \fi
1125 \ifdefined\eql@tagpos@continuous
1126 \eql@tagpos@single@eval@continuous
1127 \fi
1128 \else
1129 \eql@tagpos@row@z@
1130 \fi
1131 \eql@tagpos@prevrow@z@
1132 \eql@tagpos@headroom@z@
1133 \eql@tagpos@footroom@z@
1134 }

```

TODO: describe

```

1135 \def\eql@tagpos@single@eval@continuous{%
1136 \ifnum\eql@tagpos@row@>z@
1137 \eql@tagpos@target@eql@tagpos@shift@
1138 \else
1139 \eql@tagpos@target@dimexpr\eql@line@height@
1140 -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@relax
1141 \fi
1142 \eql@tagpos@row@one
1143 \ifdim\ifdim\eql@tagpos@target@<z@-\fi
1144 \eql@tagpos@target@<\glueexpr\eql@tagpos@snaprelax
1145 \eql@tagpos@target@z@
1146 \fi
1147 }

```

TODO: describe

```

1148 \def\eql@tagpos@adjust@eval{%
1149 \if@eqnsw
1150 \csname eql@tagpos@eval@eql@numbering@mode\endcsname
1151 \ifnum\eql@tagpos@row@>\eql@totalrows@
1152 \eql@tagpos@row@\eql@totalrows@
1153 \fi
1154 \ifdefined\eql@tagpos@doconvert
1155 \let\eql@tagpos@continuous\eql@true
1156 \fi
1157 \ifdefined\eql@tagpos@continuous
1158 \ifnum\eql@tagpos@row@>z@
1159 \eql@tagpos@adjust@eval@convert
1160 \fi
1161 \eql@tagpos@adjust@eval@continuous
1162 \fi
1163 \else
1164 \eql@tagpos@row@z@
1165 \eql@tagpos@prevrow@z@
1166 \fi
1167 }

```

TODO: describe

```

1168 \def\eql@tagpos@adjust@eval@convert{%
1169 \eql@tagpos@current@z@
1170 \eql@dimensions@for{%
1171 \ifnum\eql@row@<\eql@tagpos@row@
1172 \advance\eql@tagpos@current@dimexpr\eql@line@interline@
1173 +\eql@line@height@+\eql@line@depth@relax
1174 \fi
1175 \ifnum\eql@row@=\eql@tagpos@row@

```

```

1176     \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1177     +\eql@line@height@-\eql@tagheight@block@\relax
1178   \fi
1179 }%
1180 }

```

TODO: describe

```

1181 \def\eql@tagpos@adjust@eval@continuous{%
1182   \dimen@\dimexpr\eql@tagpos@current@-\eql@tagpos@shift@\relax
1183   \eql@tagpos@row@\eql@totalrows@
1184   \eql@tagpos@prevrow@\z@
1185   \eql@tagpos@headroom@\z@
1186   \eql@tagpos@footroom@\z@
1187   \eql@dimensions@for{%
1188     \ifnum\eql@tagpos@row@=\eql@totalrows@
1189       \eql@tagpos@headroom@\eql@line@interline@
1190       \eql@tagpos@target@\dimexpr\eql@line@interline@
1191       +\eql@line@height@-\dimen@-\eql@tagheight@block@\relax
1192       \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1193       \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1194       \advance\dimen@\eql@tagpos@target@
1195       \eql@tagpos@target@\z@
1196     \fi
1197     \ifdim\eql@tagpos@target@>%
1198       \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1199     \eql@tagpos@row@\eql@row@
1200     \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1201   \fi
1202   \advance\dimen@-\dimexpr\eql@line@interline@
1203   +\eql@line@depth@+\eql@line@height@\relax
1204   \fi
1205   \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1206     \eql@tagpos@footroom@\eql@line@interline@
1207   \fi
1208 }%
1209 }

```

TODO: describe

```

1210 \def\eql@tagpos@print@line@eval{%
1211   \ifdefined\eql@tagpos@continuous
1212     \eql@tagpos@print@line@eval@continuous
1213   \else
1214     \eql@tagpos@print@line@eval@discrete
1215   \fi
1216 }

```

TODO: describe

```

1217 \def\eql@tagpos@print@line@eval@continuous{%
1218   \if@eqnsw
1219     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1220     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1221     \eql@tagpos@plain@\eql@tagpos@target@
1222     \@tempdima\dimexpr\eql@line@height@+\eql@tagpos@headroom@
1223     -\ht\eql@tagbox@\relax
1224     \@tempdimb\dimexpr-\eql@line@depth@-\eql@tagpos@footroom@
1225     +\dp\eql@tagbox@\relax
1226     \ifnum\eql@row@=\@ne
1227       \@tempdima.5\maxdimen

```

```

1228 \fi
1229 \ifnum\eq\row@=\eq\totalrows@
1230 \tempdimb-.5\maxdimen
1231 \fi
1232 \ifdim\eq\tagpos@plain@>\tempdima
1233 \ifdim\eq\tagpos@plain@>\tempdimb
1234 \ifdim\tempdima>\tempdimb
1235 \eq\tagpos@plain@\tempdima
1236 \else
1237 \eq\tagpos@plain@\tempdimb
1238 \fi
1239 \fi
1240 \else
1241 \ifdim\eq\tagpos@plain@<\tempdimb
1242 \ifdim\tempdima>\tempdimb
1243 \eq\tagpos@plain@\tempdimb
1244 \else
1245 \eq\tagpos@plain@\tempdima
1246 \fi
1247 \fi
1248 \fi
1249 \ifnum\eq\tagpos@prevrow@>z@
1250 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1251 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1252 \eq\tagpos@raised@\eq\tagpos@plain@
1253 \let\eq\tagpos@reserve\eq>false
1254 \fi
1255 \else
1256 \ifdim\eq\tagpos@target@>%
1257 \ifdefined\eq\tagleft-1sp\relax\else z@\fi
1258 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1259 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1260 \eq\tagpos@raised@\eq\tagpos@plain@
1261 \let\eq\tagpos@reserve\eq>false
1262 \fi
1263 \else
1264 \eq\tagpos@raised@\dimexpr-\eq\line@depth@
1265 -\ht\eq\tagbox@\relax
1266 \ifdim\eq\tagpos@raised@<\eq\tagpos@plain@\else
1267 \eq\tagpos@raised@\eq\tagpos@plain@
1268 \let\eq\tagpos@reserve\eq>false
1269 \fi
1270 \fi
1271 \fi
1272 \else
1273 \ifnum\eq\tagpos@prevrow@=\eq\row@
1274 \eq\tagwidth@\eq\tagwidth@block@
1275 \else
1276 \let\eq\tagpos@reserve\eq>false
1277 \fi
1278 \fi
1279 }

```

TODO: describe

```

1280 \def\eq\tagpos@print@line@eval@discrete{%
1281 \if@eqnsw
1282 \ht\eq\tagbox@\dimexpr\ht\eq\tagbox@-\eq\tagpos@smashup@\relax
1283 \dp\eq\tagbox@\dimexpr\dp\eq\tagbox@-\eq\tagpos@smashdown@\relax

```

```

1284 \eql@tagpos@plain@\eql@tagpos@shift@
1285 \eql@tagpos@headroom@\z@
1286 \eql@tagpos@footroom@\z@
1287 \ifdim\eql@tagpos@shift@>%
1288 \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1289 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1290 \else
1291 \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1292 \fi
1293 \else
1294 \let\eql@tagpos@reserve\eql@false
1295 \fi
1296 }

```

TODO: describe

```

1297 \def\eql@tagpos@print@line@end{%
1298 \ifdefined\eql@tagpos@continuous
1299 \ifnum\eql@tagpos@prevrow@=\eql@row@
1300 \ifdefined\eql@tagpos@reserve
1301 \global\eql@appendexpand\eql@tags@container@block{%
1302 \advance\eql@tagpos@headroom@\the\dimexpr\eql@line@height@
1303 +\eql@line@depth@\relax\relax}%
1304 \eql@displaybreak@star\@M
1305 \fi
1306 \fi
1307 \fi
1308 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1309 \let\eql@SK@loaded\eql@false
1310 \let\eql@SK@label\@gobble
1311 \let\eql@SK@clearlabel\@empty
1312 \let\eql@SK@setlabel\@gobble
1313 \let\eql@SK@printlabel@right\@empty
1314 \let\eql@SK@printlabel@left\@empty
1315 \let\eql@SK@printlabel@line\@empty
1316 \def\eql@label@clean{\eql@label@org}
1317 \AddToHook{package/showkeys/after}{
1318 \let\eql@SK@loaded\eql@true
1319 \def\eql@SK@label#1{\SK@\SK@@label#1}
1320 \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1321 \eql@SK@clearlabel
1322 \def\eql@SK@@label#1>#2\SK@{%
1323 \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1324 }
1325 \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1326 \def\eql@SK@printlabel@right{%
1327 \ifx\eql@SK@lab\relax\else
1328 \rlap{\kern\marginparsep\eql@SK@lab}%
1329 \eql@SK@clearlabel
1330 \fi
1331 }
1332 \def\eql@SK@printlabel@left{%
1333 \ifx\eql@SK@lab\relax\else

```

```

1334     \llap{\eq@SK@lab\kern\marginparsep}%
1335     \eq@SK@clearlabel
1336   \fi
1337 }
1338 \def\eq@SK@printlabel@line{%
1339   \ifx\eq@SK@lab\relax\else
1340     \dimen@\prevdepth
1341     \nointerlineskip
1342     \ifdefined\eq@tagsleft
1343       \llap{%
1344         \eq@SK@lab
1345         \kern\marginparsep
1346       }%
1347     \eq@SK@clearlabel
1348   \else
1349     \rlap{%
1350       \dimen@\displaywidth
1351       \advance\dimen@\marginparsep
1352       \kern\dimen@
1353       \eq@SK@lab
1354     }%
1355   \fi
1356   \eq@SK@clearlabel
1357   \prevdepth\dimen@
1358 \fi
1359 }
1360 \let\eq@label@org\label
1361 \def\eq@label@clean{\let\SK@\@gobbletwo\eq@label@org}
1362 }

```

Labels.

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

```

1363 \def\eq@composetag@label{%
1364   \eq@SK@clearlabel
1365   \ifdefined\eq@tags@label
1366     \eq@SK@setlabel\eq@tags@label
1367   \ifdefined\eq@tags@name
1368     \let\@currentlabelname\eq@tags@name
1369   \else
1370     \let\@currentlabelname\eq@tags@name@generic
1371   \fi
1372   \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1373 \fi
1374 }

```

TODO: describe

```

1375 \def\eq@tags@printsubeqlabel{%
1376   \eq@tags@container@parent
1377   \ifdefined\eq@tags@label
1378     \eq@tags@makeblockanchor
1379     \eq@SK@setlabel\eq@tags@label
1380     \begingroup
1381     \def\@currentcounter{equation}%
1382     \eq@tags@anchor
1383     \let\@currentlabelname\eq@tags@name@generic
1384     \protected@edef\@currentlabel{\p@equation\theparentequation}%

```

```

1385     \expandafter\eq\@label@clean\expandafter{\eq\@tags@label}%
1386     \endgroup
1387     \eq\@SK@printlabel@line
1388 \fi
1389 }

```

Hyperref Anchors. **TODO:** describe

```

1390 \let\eq\Hy@anchor\@gobble
1391 \AddToHook{package/hyperref/after}{
1392   \def\eq\Hy@anchor#1{%
1393     \Hy@raisedlink{\hyper@anchor{#1}}%
1394   }%
1395 }

```

TODO: describe

```

1396 \def\eq\@tags@makeblockanchor{%
1397   \eq\@tags@glabel@step
1398   \eq\Hy@anchor\eq\@tags@glabel
1399   \edef\eq\@tags@anchor{%
1400     \def\noexpand\thepage{\thepage}%
1401     \def\noexpand\@currentHref{\eq\@tags@glabel}%
1402   }%
1403 }

```

TODO: describe

eq\@composetag@anchor

```

1404 \def\eq\@composetag@anchor{%
1405   \ifdefined\eq\@tags@tag
1406     \def\@currentcounter{equation}%
1407     \ifdefined\eq\@tags@ref
1408       \let\@currentlabel\eq\@tags@ref
1409     \else
1410       \protected@edef\@currentlabel{\p@equation\eq\@tags@tag}%
1411     \fi
1412     \eq\@tags@glabel@step
1413     \edef\@currentHref{\eq\@tags@glabel}%
1414     \eq\Hy@anchor\@currentHref
1415   \else
1416     \refstepcounter{equation}%
1417     \protected@edef\eq\@tags@tag{\theequation}%
1418   \fi
1419   \eq\@tags@anchor
1420 }

```

Tag Layout. **TODO:** describe

```

1421 \def\eq\@tags@taglayout@set@direct#1{%
1422   \def\eq\@tags@taglayout##1{#1}%
1423 }
1424 \def\eq\@tags@taglayout@set#1{%
1425   \def\eq\@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1426 }

```

TODO: describe


```

1427 \def\eql@tags@tagform@set@direct#1{%
1428   \def\eql@tags@tagform##1{#1}%
1429 }
1430 \def\eql@tags@tagform@set#1#2#3{%
1431   \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1432 }

1433 \eql@tags@taglayout@set{#1}
1434 \eql@tags@tagform@set({#1})
1435 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1436 \protected\def\tagform{\eql@tags@tagform}
1437 \protected\def\tagbox{\eql@tags@taglayout}
1438 \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:

```

1439 \protected\def\eqleqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1440 \def\eql@composetag@tag{%
1441   \eql@tagging@tagbegin
1442   \eql@tags@frame@cmd{%
1443     \eql@tags@taglayout{%
1444       \eql@tags@tagform\eql@tags@tag
1445       \eql@tagging@tagsave
1446     }%
1447   }%
1448   \eql@tagging@tagend
1449 }

```

5.7 Tag Composition

TODO: describe

```

1450 \def\eql@composetag@measure{%
1451   \ifdefined\eql@tags@tag\else
1452     \stepcounter{equation}%
1453     \let\eql@tags@tag\theequation
1454   \fi
1455   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1456   \ifdefined\eql@numbering@multi
1457     \global\let\eql@tags@container\eql@tags@container@clear
1458   \fi
1459 }

```

TODO: describe

```

1460 \def\eql@composetag@print{%
1461   \eql@composetag@anchor
1462   \eql@composetag@label
1463   \ifdefined\eql@tags@left
1464     \eqleqref{printlabel@left}
1465     \eql@composetag@tag
1466   \else
1467     \eql@composetag@tag
1468     \eqleqref{printlabel@right}

```

```

1469 \fi
1470 \global\let\eql@tags@container\eql@tags@container@clear
1471 }

```

TODO: describe

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

```

1472 \def\eql@tagbox@make#1{%
1473   \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
1474   \eql@tagwidth@\wd\eql@tagbox@
1475   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1476     \eql@tagwidth@\eql@tagwidthmin@
1477   \fi
1478   \advance\eql@tagwidth@\eql@tagsepmin@
1479 }

```

TODO: describe

```

1480 \def\eql@tagbox@print@adjustheadroom{%
1481   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1482   \ifdim\dimen@>\z@
1483     \ifdim\dimen@>\eql@tagpos@headroom@
1484       \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1485     \else
1486       \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1487     \fi
1488   \fi
1489 }

```

TODO: describe

```

1490 \def\eql@tagbox@print@adjustfootroom{%
1491   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1492   \ifdim\dimen@>\z@
1493     \ifdim\dimen@>\eql@tagpos@footroom@
1494       \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1495     \else
1496       \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1497     \fi
1498   \fi
1499 }

```

TODO: describe

```

1500 \def\eql@tagbox@print@extendabove{%
1501   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1502   \ifdim\dimen@>\z@
1503     \global\eql@appendexpand\eql@display@container{%
1504       \eql@display@aboveextend@the\dimen@\relax}%
1505   \fi
1506 }

```

TODO: describe

```

1507 \def\eql@tagbox@print@extendbelow{%
1508   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1509   \ifdim\dimen@>\z@
1510     \global\eql@appendexpand\eql@display@container{%
1511       \eql@display@belowextend@the\dimexpr\dimen@\relax}%
1512   \fi
1513 }

```

TODO: describe

```
1514 \def\eql@tagbox@print@prepare{%
1515   \ifdefined\eql@tagpos@reserve
1516     \eql@tagpos@current@\eql@tagpos@plain@
1517   \else
1518     \eql@tagpos@current@\eql@tagpos@raised@
1519   \fi
1520   \ifdim\eql@tagpos@headroom@>\z@
1521     \eql@tagbox@print@adjustheadroom
1522   \fi
1523   \ifdim\eql@tagpos@footroom@>\z@
1524     \eql@tagbox@print@adjustfootroom
1525   \fi
1526   \ifnum\eql@row@=\@ne
1527     \eql@tagbox@print@extendabove
1528   \fi
1529   \ifnum\eql@row@=\eql@totalrows@
1530     \eql@tagbox@print@extendbelow
1531   \fi
1532 }
```

TODO: describe

```
1533 \def\eql@tagbox@print@tagsright{%
1534   \eql@tagbox@print@prepare
1535   \kern-\wd\eql@tagbox@
1536   \raise\eql@tagpos@current@\box\eql@tagbox@
1537 }
```

TODO: describe

```
1538 \def\eql@tagbox@print@tagsleft{%
1539   \eql@display@firstavail@set\z@
1540   \eql@tagbox@print@prepare
1541   \wd\eql@tagbox@\z@
1542   \raise\eql@tagpos@current@\box\eql@tagbox@
1543 }
```

$\eql@tagbox@print@cell$

```
1544 \def\eql@tagbox@print@cell{%
1545   \eql@tagging@tagaddbox
1546   \ifdefined\eql@tagsleft
1547     \ifdefined\eql@tagpos@reserve
1548       \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1549         \let\eql@tagpos@reserve\eql@false
1550       \fi
1551     \fi
1552     \if@eqnsw
1553       \eql@tagbox@print@tagsleft
1554     \fi
1555     \kern\displaywidth
1556   \else
1557     \kern\displaywidth
1558     \ifdefined\eql@tagpos@reserve
1559       \ifdim\eql@tagwidth@>%
1560         \dimexpr\displaywidth-\eql@line@width@+\eql@tagfuzz@\relax
1561       \let\eql@tagpos@reserve\eql@false
1562     \fi
```

```

1563 \fi
1564 \if@eqnsw
1565 \eq@tagbox@print@tagsright
1566 \fi
1567 \fi
1568 }

```

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:

```

1569 \eq@amsmath@undefine\c@parentequation
1570 \eq@amsmath@undefine\theparentequation
1571 \ifdefined\c@parentequation\else
1572 \newcounter{parentequation}
1573 \fi

```

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

```

1574 \def\eq@subequations@template{\theparentequation\alph{equation}}

```

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

```

1575 \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`:

```

1576 \def\eq@subequations@init{%
1577 \edef\eq@subequations@restorecounter{%
1578 \global\c@equation\the\c@equation\relax}%
1579 \eq@tags@container@block
1580 \ifdefined\eq@tags@tag
1581 \eq@tags@glabel@step
1582 \protected@edef\theHparentequation{\eq@tags@glabel}%
1583 \protected@edef\theparentequation{\eq@tags@tag}%
1584 \else
1585 \advance\c@equation\@ne
1586 \protected@edef\theparentequation{\theequation}%
1587 \ifdefined\theHequation
1588 \protected@edef\theHparentequation{\theHequation}%
1589 \fi

```

```

1590 \fi
1591 \global\c@parentequation\c@equation
1592 \global\c@equation\z@
1593 \let\theequation\eq@subequations@template
1594 \def\theHequation{\theHparentequation.\arabic{equation}}%
1595 }

```

1@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:

```

1596 \def\eq@subequations@close{%
1597   \ifnum\c@equation=\z@
1598     \eq@subequations@restorecounter
1599   \else
1600     \global\c@equation\c@parentequation
1601   \fi
1602 }

```

6.2 Environment

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

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

```

1603 \def\eq@subequations@start{%
1604   \let\eq@tags@container@block\eq@tags@container@clear
1605   \eq@nextopt@process{subequations}%
1606   \eq@subequations@init
1607   \eq@tags@glabel@step
1608   \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1609   \eq@Hy@anchor\eq@subequations@currentHref
1610   \edef\eq@subequations@thepage{\thepage}%
1611   \def\@currentcounter{equation}%
1612   \let\@currentHref\eq@subequations@currentHref
1613   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1614   \eq@tags@container@block
1615   \ifdefined\eq@tags@name
1616     \let\@currentlabelname\eq@tags@name
1617   \else
1618     \let\@currentlabelname\eq@tags@name@generic
1619   \fi
1620   \let\eq@subequations@active\eq@true
1621   \ifdefined\eq@tags@label
1622     \eq@SK@label\eq@tags@label
1623   \fi
1624   \ignorespaces
1625 }

```

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

```

1626 \def\eq@subequations@end{%
1627   \ifnum\c@equation>\z@
1628     \eq@tags@container@block
1629     \ifdefined\eq@tags@label
1630       \begingroup

```

```

1631      \def\@currentcounter{equation}%
1632      \let\thepage\eq\@subequations\thepage
1633      \let\@currentHref\eq\@subequations\@currentHref
1634 % \TODO how about tag* ?! also within equations!
1635      \protected@edef\@currentlabel{\p@equation\theparentequation}%
1636      \ifdefined\eq\@tags@name
1637          \let\@currentlabelname\eq\@tags@name
1638      \else
1639          \let\@currentlabelname\eq\@tags@name@generic
1640      \fi
1641      \expandafter\eq\@label@clean\expandafter{\eq\@tags@label}%
1642      \endgroup
1643      \fi
1644  \fi
1645  \eq\@subequations@close
1646 }

```

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

```

1647 \newenvironment{eq\@subequations}{%
1648   \eq\@verbose@info\eq\@verbose@msg@enterenv
1649   \eq\@subequations@testall\eq\@subequations@start%
1650 }{%
1651   \eq\@subequations@end
1652   \ignorespacesafterend
1653   \eq\@verbose@info\eq\@verbose@msg@leaveenv
1654 }

```

TODO: describe

```

1655 \def\eq\@subequations@testall{\eq\@parseopt\eq\@subequations@parseopt}
1656 \def\eq\@subequations@parseopt{%
1657   \ifx\eq\@parseopt@token[%]
1658     \let\eq\@parseopt@next\eq\@parseopt@opt
1659   \fi
1660   \ifx\eq\@parseopt@token\eq\@atxi
1661     \let\eq\@parseopt@next\eq\@parseopt@label
1662   \fi
1663   \ifx\eq\@parseopt@token\eq\@atxii
1664     \let\eq\@parseopt@next\eq\@parseopt@label
1665   \fi
1666   \ifx\eq\@parseopt@token\label
1667     \let\eq\@parseopt@next\eq\@parseopt@end
1668   \fi
1669 }

```

6.3 Subequation Scheme

TODO: describe

```

1670 \def\eq\@numbering@subeq@init{%
1671   \let\eq\@save@theequation\theequation
1672   \let\eq\@save@theHequation\theHequation
1673   \eq\@subequations@init
1674   \let\eq\@tags@container@parent\eq\@tags@container@block
1675   \let\eq\@tags@container@block\eq\@tags@container@clear
1676 }

```

TODO: describe

```
1677 \def\eq@numbering@subeq@test{%
1678   \ifnum\eq@tagrows@<\tw@
1679     \let\eq@tags@container@block\eq@tags@container@parent
1680     \let\eq@numbering@subeq@use\eq@false
1681     \let\theequation\eq@save@theequation
1682     \let\theHequation\eq@save@theHequation
1683     \eq@subequations@restorecounter
1684   \fi
1685 }
```

TODO: describe

```
1686 % \TODO note must not use setcounter here (when calc is loaded)
1687 \def\eq@numbering@subeq@close{%
1688   \eq@subequations@close
1689 }
```

7 Display Equations Support

TODO: describe

```
1690 \let\eq@display@injectbefore\@undefined
1691 \let\eq@display@injectafter\@undefined
1692 \let\eq@interline@container\@undefined
1693 \def\eq@interline@container@clear{%
1694   \eq@displaybreak@pen@\@MM
1695   \eq@vspaceskip@\z@skip
1696 }
```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1697 \interdisplaylinepenalty\@M
```

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

```
1698 \def\eq@getdsp@pen#1{%
1699   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1700 }
```

TODO: allow a displaybreak before equations

```
1701 \protected\def\eq@displaybreak@default{%
1702   \eq@warning{Invalid use of \string\displaybreak}{}}%
1703   \eq@teststaroropt@loose\@gobble\eq@gobbleopt{}}
1704 \eq@amsmath@after{\let\eq@displaybreak@default\displaybreak}
1705 \eq@amsmath@let\displaybreak\eq@displaybreak@default

1706 \newcount\eq@displaybreak@pen@
1707 \newcount\eq@displaybreak@prepen@
1708 \newcount\eq@displaybreak@postpen@
```

TODO: describe

```
1709 \protected\def\eqldisplaybreak{%
1710   \relax
1711   \eql@ampprotecttwo\eql@teststaroropt@tight
1712   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1713 }

1714 \def\eqldisplaybreak@star#1{%
1715   \global\eql@appendexpand\eql@interline@container{%
1716     \eqldisplaybreak@pen@\the\numexpr#1\relax\relax}%
1717 }

1718 \def\eqldisplaybreak@level[#1]{%
1719   \ifnum#1<\z@
1720     \global\eql@append\eql@interline@container{\eqldisplaybreak@pen@\@MM}%
1721   \else
1722     \global\eql@appendexpand\eql@interline@container{%
1723       \eqldisplaybreak@pen@-\@getpen{#1}\relax}%
1724   \fi
1725 }
```

TODO: describe

```
1726 \def\eqldisplaybreak@pre#1{%
1727   \ifnum#1<\z@
1728     \eqldisplaybreak@prepen@\@MM
1729   \else
1730     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1731   \fi
1732 }
```

TODO: describe

```
1733 \def\eqldisplaybreak@post#1{%
1734   \ifnum#1<\z@
1735     \eqldisplaybreak@postpen@\@MM
1736   \else
1737     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1738   \fi
1739 }
```

TODO: describe

```
1740 \def\eqldisplaybreak@inter#1{%
1741   \ifnum#1<\z@
1742     \interdisplaylinepenalty\@M
1743   \else
1744     \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax
1745   \fi
1746 }
```

7.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```
1747 \newskip\eqlvspaceskip@
```



```

1748 \let\eql@vspace@org\vspace
1749 \def\eql@vspace{%
1750   \ifvmode
1751     \expandafter\eql@vspace@immediate
1752   \else
1753     \expandafter\eql@vspace@line
1754   \fi
1755 }

```

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

```

1756 \def\eql@vspace@line{%
1757   \eql@ifstar@loose\eql@vspace@addfixedbefore\eql@vspace@add
1758 }
1759 \def\eql@vspace@add#1{%
1760   \global\eql@appendexpand\eql@interline@container{%
1761     \advance\eql@vspaceskip@\the\glueexpr#1\relax\relax}}
1762 \def\eql@vspace@addfixedbefore#1{%
1763   \global\eql@appendexpand\eql@interline@container{%
1764     \noexpand\eql@append\noexpand\eql@display@injectbefore{%
1765       \skip@\the\glueexpr#1\relax\relax
1766       \penalty\@M
1767       \vskip\skip@
1768     \global\advance\eql@line@interline@\skip@
1769   }%
1770 }%
1771 }
1772 \def\eql@vspace@addfixedafter#1{%
1773   \global\eql@appendexpand\eql@interline@container{%
1774     \noexpand\eql@append\noexpand\eql@display@injectafter{%
1775       \dimen@\prevdepth
1776       \hrule\@height\z@
1777       \skip@\the\glueexpr#1\relax\relax
1778       \penalty\@M
1779       \vskip\skip@
1780     \global\advance\eql@line@interline@\skip@
1781     \prevdepth\dimen@
1782   }%
1783 }%
1784 }

```

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

```

1785 \def\eql@vspace@immediate{%
1786   \noalign\bgroup
1787   \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
1788 }
1789 \def\eql@vspace@fixed#1{%
1790   \skip@\glueexpr#1\relax
1791   \ifnum\eql@row@=\@ne
1792     \global\eql@appendexpand\eql@display@container{%
1793       \advance\eql@abovespace@\the\skip@\relax}%
1794   \else\ifnum\eql@row@>\eql@totalrows@
1795     \global\eql@appendexpand\eql@display@container{%
1796       \advance\eql@belowspace@\the\skip@\relax}%
1797   \else
1798     \dimen@\prevdepth
1799     \hrule\@height\z@
1800     \penalty\@M

```

```

1801     \vskip\skip@
1802     \global\advance\eql@line@interline@\skip@
1803     \prevdepth\dimen@
1804   \fi\fi
1805 \egroup
1806 }
1807 \def\eql@vspace@discardable#1{%
1808   \skip@\glueexpr#1\relax
1809   \ifnum\eql@row@=\@ne
1810     \global\eql@appendexpand\eql@display@container{%
1811       \advance\eql@abovespace@\the\skip@\relax}%
1812   \else\ifnum\eql@row@>\eql@totalrows@
1813     \global\eql@appendexpand\eql@display@container{%
1814       \advance\eql@belowspace@\the\skip@\relax}%
1815   \else
1816     \vskip\skip@
1817     \global\advance\eql@line@interline@\skip@
1818   \fi\fi
1819 \egroup
1820 }

```

7.3 Default Vertical Spacing

TODO: describe

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

```

1821 \newbox\eql@strutbox@
1822 \def\eql@strut@depth{.3}
1823 \def\eql@strut{\copy\eql@strutbox@}
1824 \let\eql@strut@cell\eql@strut
1825 \let\eql@strut@tag\eql@strut
1826 \def\eql@strut@make{%
1827   \setbox\eql@strutbox@\hbox{%
1828     \@tempdimb\dimexpr
1829       \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1830     \@tempdima\dimexpr
1831       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
1832     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1833   }
1834 }
1835 \AtBeginDocument{\eql@strut@make}

```

TODO: describe

```

1836 \def\eql@spread@set{%
1837   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
1838   +\normalbaselineskip-\baselineskip\relax
1839   \ifdim\eql@spread@>\z@
1840     \openup\eql@spread@
1841     \ifdefined\spread@equation
1842       \let\spread@equation\@empty
1843     \fi
1844   \fi
1845 }

```

7.4 Entry and Exit

TODO: describe

```
1846 \let\eql@beamerbasecolor@fix@empty
1847 \AddToHook{package/beamerbasecolor/after}{%
1848   \def\eql@beamerbasecolor@fix{%
1849     \donotcolorouterdisplaymaths
1850     \donotcoloroutermaths
1851     \beamer@setdisplaymathcolor
1852   }%
1853 }
```

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

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

```
1854 \newskip\eql@abovespace@
1855 \newskip\eql@belowspace@
```

`\eql@display@enter`

```
1856 \def\eql@display@enter{%
1857   \if@noskipsec\leavevmode\par\fi
1858   \ifvmode
1859     \eql@prevdepth@\prevdepth
1860     \nointerlineskip
1861     \noindent
1862   \else
1863     \eql@prevdepth@\maxdimen
1864   \fi
1865   \eql@beamerbasecolor@fix
1866 }
```

`\eql@display@adjust`

```
1867 \def\eql@display@adjust{%
1868   \ifdefined\eql@display@linewidth
1869     \displaywidth\glueexpr\eql@display@linewidth\relax
1870     \advance\displaywidth-\displayindent
1871   \fi
1872   \ifdefined\eql@display@marginleft
1873     \advance\displaywidth\displayindent
1874     \displayindent\glueexpr\eql@display@marginleft\relax
1875     \advance\displaywidth-\displayindent
1876   \fi
1877   \ifdefined\eql@display@marginright
1878     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1879   \fi
1880   \ifdim\displaywidth<\z@
1881     \displaywidth\z@
1882   \fi
1883 }
```

`\eql@display@init`

```
1884 \def\eql@display@init{%
1885   \let\displaybreak\eql@displaybreak
1886   \let\eql@vspace@org\vspace
1887   \let\vspace\eql@vspace
1888   \let\eqncontrol\eql@control
```

```

1889 \let\eqldisplay@injectbefore\@empty
1890 \let\eqldisplay@injectafter\@empty
1891 \eql@spread@set
1892 \eql@strut@make
1893 \let\eql@frame@cmd\@undefined
1894 }

```

\eqldisplay@print

```

1895 \def\eqldisplay@print{%
1896 \let\eqldisplay@container\@empty
1897 \eqldisplay@firstavail@z@
1898 \eqldisplay@aboveextend@z@
1899 \eqldisplay@belowextend@z@
1900 \global\let\eql@interline@container\eql@interline@container@clear
1901 }

```

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

```

1902 \def\eqldisplay@halign@init#1{%
1903 \eql@row@z@
1904 \eql@prevgraf@\prevgraf
1905 \everycr{\noalign{%
1906 \global\advance\eql@row@\@ne
1907 \prevgraf\numexpr\prevgraf+\@ne\relax
1908 #1%
1909 }}%
1910 }

```

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

```

1911 \def\eqldisplay@halign@start{%
1912 \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1913 \ifdim\eql@prevdepth=\maxdimen\else
1914 \prevdepth\eql@prevdepth@
1915 \fi
1916 \ifdim\prevdepth=-\@m\p@else
1917 \ifdefined\eqldisplay@height
1918 \skip@\baselineskip
1919 \advance\skip@-\glueexpr\eqldisplay@height\relax
1920 \advance\skip@-\prevdepth\relax
1921 \ifdim\skip@<\lineskiplimit
1922 \skip@\lineskip
1923 \fi
1924 \advance\skip@-\eql@spread@\relax
1925 \vskip\skip@
1926 \nointerlineskip
1927 \else
1928 \vskip-\eql@spread@\relax
1929 \fi
1930 \fi
1931 }

```

TODO: describe

```

1932 \def\eqldisplay@vspace{%
1933 \advance\abovedisplayskip\eql@abovespace@
1934 \advance\belowdisplayskip\eql@belowspace@
1935 }

```

TODO: describe

```
1936 \def\eqldisplay@vspace@native{%
1937   \advance\abovedisplayskip\eql@abovespace@
1938   \advance\belowdisplayskip\eql@belowspace@
1939   \advance\abovedisplayshortskip\eql@abovespace@
1940   \advance\belowdisplayshortskip\eql@belowspace@
1941 }
```

TODO: describe

```
1942 \def\eqldisplay@penalty{%
1943   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1944     \postdisplaypenalty\eqldisplaybreak@postpen@
1945   \fi
1946   \ifnum\eqldisplaybreak@pen@=\@MM\else
1947     \postdisplaypenalty\eqldisplaybreak@pen@
1948   \fi
1949   \ifnum\eqldisplaybreak@prepen@=\@MM\else
1950     \predisplayskip\eqldisplaybreak@prepen@
1951   \fi
1952 }
```

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

```
1953 \def\eqldisplay@halign@end{%
1954   \eql@interline@container
1955   \eqldisplay@injectbefore
1956   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
1957   \ifdefined\eqldisplay@depth
1958     \prevdepth\glueexpr\eqldisplay@depth\relax
1959   \fi
1960 }
```

`\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?

```
1961 \def\eqldisplay@close{%
1962   \eqldisplay@container
1963   \ifdim\eqldisplay@firstavail@<\z@
1964     \eqldisplay@firstavail@\z@
1965   \fi
1966   \eql@skip@mode@leave@\z@
1967   \ifdim\eql@prevdepth@=\maxdimen
1968     \ifdim\predisplaysize=-\maxdimen
1969       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1970       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1971     \else
1972       \eql@skip@mode@above@\z@
1973       \eql@skip@mode@below@\z@
1974       \advance\eqldisplay@firstavail@\displayindent
1975       \ifdim\eqldisplay@firstavail@>\predisplaysize
1976         \ifcase\eql@skip@mode@short\relax
1977           \or
1978             \eql@skip@mode@above@\@ne
1979           \or
1980             \eql@skip@mode@above@\@ne
1981           \ifnum\eql@totalrows@=\@ne
1982             \eql@skip@mode@below@\@ne
```

```

1983         \fi
1984     \or
1985         \eqL@skip@mode@above@ \@ne
1986         \eqL@skip@mode@below@ \@ne
1987     \fi
1988 \fi
1989 \fi
1990 \else
1991     \ifdim\eqL@prevdepth@=-\@m\p@
1992         \eqL@skip@mode@above@\eqL@skip@mode@top@above\relax
1993         \eqL@skip@mode@below@\eqL@skip@mode@top@below\relax
1994     \else
1995         \eqL@skip@mode@above@\eqL@skip@mode@par@above\relax
1996         \eqL@skip@mode@below@\eqL@skip@mode@par@below\relax
1997     \fi
1998 \fi
1999 \ifcase\eqL@skip@mode@above@
2000 \or\or\or
2001     \predisplaypenalty\z@
2002 \or
2003     \predisplaypenalty\z@
2004 \fi
2005 \ifcase\eqL@skip@mode@below@
2006 \or\or\or
2007     \eqL@skip@mode@leave@ \@ne
2008 \or
2009     \eqL@skip@mode@leave@ \tw@
2010 \fi
2011 \ifdefined\eqL@skip@force@above
2012     \eqL@skip@mode@above@\eqL@skip@force@above\relax
2013 \fi
2014 \ifdefined\eqL@skip@force@below
2015     \eqL@skip@mode@below@\eqL@skip@force@below\relax
2016 \fi
2017 \ifdefined\eqL@skip@force@leave
2018     \eqL@skip@mode@leave@\eqL@skip@force@leave\relax
2019 \fi
2020 \ifnum\eqL@skip@mode@leave@>\z@
2021     \postdisplaypenalty\z@
2022 \fi
2023 \ifcase\eqL@skip@mode@above@
2024     \abovedisplayskip\glueexpr\eqL@skip@long@above\relax
2025 \or
2026     \abovedisplayskip\glueexpr\eqL@skip@short@above\relax
2027 \or
2028     \abovedisplayskip\glueexpr\eqL@skip@cont@above\relax
2029 \or
2030     \abovedisplayskip\glueexpr\eqL@skip@par@above\relax
2031 \or
2032     \abovedisplayskip\glueexpr\eqL@skip@top@above\relax
2033 \or
2034     \abovedisplayskip\z@skip
2035 \or
2036     \abovedisplayskip\glueexpr\eqL@skip@med@above\relax
2037 \or
2038     \abovedisplayskip\glueexpr\eqL@skip@custom@above\relax
2039 \fi
2040 \ifcase\eqL@skip@mode@below@

```

```

2041 \belowdisplayskip\glueexpr\eq\@skip@long@below\relax
2042 \or
2043 \belowdisplayskip\glueexpr\eq\@skip@short@below\relax
2044 \or
2045 \belowdisplayskip\glueexpr\eq\@skip@cont@below\relax
2046 \or
2047 \belowdisplayskip\glueexpr\eq\@skip@par@below\relax
2048 \or
2049 \belowdisplayskip\glueexpr\eq\@skip@top@below\relax
2050 \or
2051 \belowdisplayskip\z@skip
2052 \or
2053 \belowdisplayskip\glueexpr\eq\@skip@med@below\relax
2054 \or
2055 \belowdisplayskip\glueexpr\eq\@skip@custom@below\relax
2056 \fi
2057 \global\eq\@skip@mode@leave@\eq\@skip@mode@leave@
2058 \eq\@interline@container
2059 \advance\eq\@belowspace@\eq\@vspaceskip@
2060 \eq\@display@penalty
2061 \eq\@display@vspace
2062 \skip@\glueexpr\eq\@skip@tag@above\relax
2063 \ifdim\skip@>\abovedisplayskip
2064 \skip@\abovedisplayskip
2065 \fi
2066 \advance\abovedisplayskip-\eq\@display@aboveextend@\relax
2067 \ifdim\abovedisplayskip<\skip@
2068 \abovedisplayskip\skip@
2069 \fi
2070 \skip@\glueexpr\eq\@skip@tag@below\relax
2071 \ifdim\skip@>\belowdisplayskip
2072 \skip@\belowdisplayskip
2073 \fi
2074 \ifdim\eq\@display@belowextend@>\z@
2075 \advance\belowdisplayskip-\eq\@display@belowextend@\relax
2076 \ifdim\belowdisplayskip<\skip@
2077 \belowdisplayskip\skip@
2078 \fi
2079 \fi
2080 }

```

TODO: describe

```

2081 \def\eq\@display@leave{%
2082 \prevgraf\eq\@prevgraf@
2083 \ifcase\eq\@skip@mode@leave@
2084 \or
2085 \endgraf
2086 \or
2087 \endgraf
2088 \prevdepth-\@m\p@
2089 \fi
2090 }

```

TODO: describe

```

2091 \def\eq\@display@nest{%
2092 \let\displaybreak\eq\@displaybreak@default
2093 \let\intertext\eq\@intertext@default
2094 \let\vspace\eq\@vspace@org

```

2095 }

TODO: describe

```
2096 \def\eqldisplay@restore{%
2097   \let\label\eqldlabel@org
2098   \let\tag\eqldtag@default
2099   \let\raisetag\eqldraisetag@default
2100   \let\displaybreak\eqldisplaybreak@default
2101   \let\intertext\eqldintertext@default
2102   \let\vspace\eqldvspace@org
2103 }
```

TODO: describe

```
2104 \eqldappend\@arrayparboxrestore{%
2105   \eqlddisplay@restore
2106   \ifdefined\eqldampproof@active
2107     \eqldamprevert
2108   \fi
2109   \@displayfalse
2110 }
```

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)

```
2111 \def\eqldstack@enable{%
2112   \let\eqldstack@save@equations\eqldstack@save@equations@
2113   \let\eqldstack@save@box\eqldstack@save@box@
2114 }
```

TODO: describe

```
2115 \let\eqldstack@save@equations\eqldstack@enable
2116 \let\eqldstack@save@box\eqldstack@enable
2117 \let\eqldstack@restore\@empty
```

TODO: describe

```
2118 \def\eqldstack@save@reg#1{\global#1\the#1\relax}
2119 \def\eqldstack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

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

\eqldprevgraf@, \eqldskip@mode@leave@, \eqldshape@lastrow, \eqldframe@prevcmd

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

```
2120 \def\eqldstack@save@equations@{%
2121   \let\eqldstack@numbering@eqnswinit\eqldnumbering@eqnswinit
2122   \let\eqldstack@cell@container\eqldcell@container
2123   \let\eqldstack@tags@container\eqldtags@container
2124   \let\eqldstack@interline@container\eqldinterline@container
2125   \let\eqldstack@block@container\eqlddisplay@container
2126   \let\eqldstack@dimensions@tab\eqlddimensions@tab
2127   \edef\eqldstack@restore{%
2128     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2129     \eqldstack@save@let\eqldstack@numbering@eqnswinit\eqldnumbering@eqnswinit
```



```

2130 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2131 \eql@stack@save@let\eql@stack@tags@container\eql@tags@container
2132 \eql@stack@save@let\eql@stack@interline@container\eql@interline@container
2133 \eql@stack@save@let\eql@stack@dimensions@tab\eql@dimensions@tab
2134 \eql@stack@save@let\eql@stack@block@container\eql@display@container
2135 \eql@stack@save@reg\eql@column@
2136 \eql@stack@save@reg\eql@totalcolumns@
2137 \eql@stack@save@reg\eql@line@avail@
2138 \eql@stack@save@reg\eql@line@pos@
2139 \eql@stack@save@reg\eql@line@width@
2140 \eql@stack@save@reg\eql@line@depth@
2141 \eql@stack@save@reg\eql@line@height@
2142 \eql@stack@save@reg\eql@line@prevdepth@
2143 \eql@stack@save@reg\eql@line@interline@
2144 \eql@stack@save@reg\eql@totalheight@
2145 \eql@stack@save@reg\eql@tagwidth@max@
2146 \eql@stack@save@reg\eql@tagpos@row@
2147 \eql@stack@save@reg\eql@row@
2148 \eql@stack@save@reg\eql@tagrows@
2149 }%
2150 }

```

TODO: describe

```

2151 \def\eql@stack@save@box@{%
2152 \let\eql@stack@cell@container\eql@cell@container
2153 \edef\eql@stack@restore{%
2154 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2155 \eql@stack@save@reg\eql@row@
2156 }%
2157 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2158 \def\eql@measure@init#1#2{%
2159 \eql@dimensions@reset
2160 \let\eql@display@container\@empty
2161 \eql@numbering@measure@init
2162 \eql@row@\z@
2163 \eql@totalheight@\z@
2164 \eql@totalrows@\@M
2165 \eql@line@prevdepth@-\@m\p@
2166 \eql@line@interline@\z@
2167 \tabskip\z@skip
2168 \everycr{\noalign{%
2169 \global\advance\eql@row@\@ne
2170 #1%
2171 }}%
2172 \global\let\eql@interline@container\eql@interline@container@clear
2173 \eql@measure@savestate
2174 \eql@display@halign@letcr{#2}%

```

2175 }

TODO: describe

```
2176 \def\eq@measure@tag{%
2177   \eq@tagwidth@z@
2178   \ifdefined\eq@numbering@multi
2179     \if@eqnsw
2180       \eq@tags@container
2181       \eq@tagbox@make\eq@composetag@measure
2182       \ifdefined\eq@tagpos@reserve\else
2183         \eq@tagwidth@z@
2184       \fi
2185     \fi
2186   \fi
2187 }
```

TODO: describe

```
2188 \def\eq@measure@endrow{%
2189   \ifdim\eq@line@prevdepth@=-\m@p\else
2190     \dimen@dimexpr\baselineskip-\eq@line@height@-\eq@line@prevdepth@relax
2191     \ifdim\dimen@<\lineskiplimit
2192       \dimen@lineskip
2193     \fi
2194     \advance\eq@line@interline@\dimen@
2195   \fi
2196   \eq@dimensions@endrow
2197   \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2198     \global\eq@tagwidth@max@\eq@tagwidth@
2199   \fi
2200   \ifdim\eq@tagwidth@>z@
2201     \global\advance\eq@tagrows@one
2202   \fi
2203   \global\advance\eq@totalheight@\dimexpr
2204     \eq@line@interline@+\eq@line@height@+\eq@line@depth@
2205   \global\eq@line@interline@z@
2206   \global\eq@line@prevdepth@\eq@line@depth@
2207 }
```

TODO: describe

```
2208 \def\eq@measure@close{%
2209   \advance\eq@row@-\tw@
2210   \eq@totalrows@\eq@row@
2211   \ifnum\eq@totalrows@>z@
2212     \eq@dimensions@get@one
2213     \eq@topheight@\dimexpr\eq@line@height@+\eq@line@interline@relax
2214     \eq@dimensions@get\eq@totalrows@
2215     \eq@bottomdepth@\eq@line@depth@
2216   \fi
2217   \eq@numbering@measure@blocktag
2218   \begingroup
2219     \eq@tags@container
2220     \if@eqnsw
2221       \eq@tagbox@make\eq@composetag@measure
2222       \ifdefined\eq@tagpos@reserve\else
2223         \eq@tagwidth@z@
2224       \fi
2225     \eq@dimensions@saveblocktag
```

```

2226     \else
2227     \eqldimensions@savenoblocktag
2228     \eqlnumbering@warnunused
2229     \fi
2230 \endgroup
2231 \eqldimensions@get\z@
2232 \eql@measure@restorestate
2233 }

```

measure@restorestate
eql@measure@savestate

```

2234 \let\eql@measure@restorestate\@empty
2235 \def\eql@measure@savestate{%
2236   \begingroup
2237   \def\@elt##1{%
2238     \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2239     \global\edef\@gtempa{\cl@ckpt}%
2240   \endgroup
2241   \let\eql@measure@restorestate\@gtempa
2242 }

```

8.2 Line Breaks

TODO: describe

\eqldisplay@cr

```

2243 \protected\def\eqldisplay@cr{\eqlsrbgroup
2244   \eql@ifnextgobble@tight~%
2245   {\numbernext\let\eql@punct@line\@empty\eqldisplay@cr@star}%
2246   \eqldisplay@cr@star
2247 }

```

\eqldisplay@cr@star

```

2248 \def\eqldisplay@cr@star{%
2249   \eql@teststaropt@tight{%
2250     \global\eql@append\eql@interline@container{\eqldisplaybreak@pen@MM}%
2251     \eqldisplay@cr@opt}%
2252   \eqldisplay@cr@opt\z@skip
2253 }

```

\eqldisplay@cr@opt

```

2254 \def\eqldisplay@cr@opt[#1]{\eqlsregroup
2255   \eqldisplay@endline
2256   \cr

2257   \noalign{%
2258     \eql@interline@container
2259     \eqldisplay@injectbefore
2260     \ifnum\eqldisplaybreak@pen@MM
2261       \penalty\interdisplaylinepenalty
2262     \else
2263       \penalty\eqldisplaybreak@pen@
2264     \fi
2265     \advance\eql@vspaceskip@glueexpr#1\relax
2266     \vskip\eql@vspaceskip@

```

```

2267 \global\advance\eqL@line@interline@\eqL@vspaceskip@
2268 \eqL@display@injectafter
2269 \global\let\eqL@interline@container\eqL@interline@container@clear
2270 }%
2271 }

```

display@halign@letcr

```

2272 \def\eqL@display@halign@letcr#1{%
2273 \let\\ \eqL@display@cr
2274 \let\eqL@display@endline#1%
2275 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2276 \def\eqL@intertext@default{\eqL@error{Invalid use of \string\intertext}}
2277 \eqL@amsmath@let\intertext\eqL@intertext@default

```

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

```

2278 \def\eqL@intertext@process{%
2279 \eqL@display@endline
2280 \cr
2281 \ifmeasuring@
2282 \expandafter\@gobble
2283 \else
2284 \expandafter\eqL@intertext@print
2285 \fi
2286 }

```

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

```

2287 \def\eqL@intertext@print#1{%
2288 \noalign{%
2289 \eqL@display@halign@end
2290 \let\eqL@skip@force@below\z@
2291 \let\eqL@skip@force@above\z@
2292 \eqL@setkeys{intertext}\eqL@intertext@opt
2293 \openup-\eqL@spread@
2294 \penalty\postdisplaypenalty
2295 \ifcase\eqL@skip@force@below\relax
2296 \advance\eqL@vspaceskip@\glueexpr\eqL@skip@long@below\relax
2297 \or
2298 \advance\eqL@vspaceskip@\glueexpr\eqL@skip@short@below\relax
2299 \or
2300 \advance\eqL@vspaceskip@\glueexpr\eqL@skip@cont@below\relax
2301 \or
2302 \advance\eqL@vspaceskip@\glueexpr\eqL@skip@par@below\relax
2303 \or
2304 \advance\eqL@vspaceskip@\glueexpr\eqL@skip@top@below\relax
2305 \or
2306 \advance\eqL@vspaceskip@\z@skip
2307 \or
2308 \advance\eqL@vspaceskip@\glueexpr\eqL@skip@med@below\relax
2309 \or

```

```

2310     \advance\eq\vsppskip@glueexpr\eq\skip@custom@below\relax
2311 \fi
2312 \vskip\eq\vsppskip@
2313 \global\let\eq\interline@container\eq\interline@container@clear
2314 \vbox{%
2315     \@parboxrestore
2316     \ifdim
2317         \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2318     \else
2319         \parshape\@ne
2320         \@totalleftmargin\linewidth
2321     \fi
2322     \noindent
2323     \prevgraf\eq\prevgraf@
2324     \ignorespaces
2325     #1%
2326     \par
2327     \global\eq\prevgraf@\prevgraf
2328 }%
2329 \penalty\predisplaypenalty
2330 \ifcase\eq\skip@force@above\relax
2331     \vskip\glueexpr\eq\skip@long@above\relax
2332 \or
2333     \vskip\glueexpr\eq\skip@short@above\relax
2334 \or
2335     \vskip\glueexpr\eq\skip@cont@above\relax
2336 \or
2337     \vskip\glueexpr\eq\skip@par@above\relax
2338 \or
2339     \vskip\glueexpr\eq\skip@top@above\relax
2340 \or
2341     \vskip\z@skip
2342 \or
2343     \vskip\glueexpr\eq\skip@med@above\relax
2344 \or
2345     \vskip\glueexpr\eq\skip@custom@above\relax
2346 \fi
2347 % \eq\prevdepth@\maxdimen
2348 \eq\prevdepth@\z@
2349 \eq\display@halign@start
2350 }
2351 }

```

TODO: describe

```

2352 \newenvironment{eq\intertext}{%
2353     \eq\testopt@tight\eq\intertext@{}%
2354 }{%
2355     \aftergroup\eq\intertext@after
2356     \ignorespacesafterend
2357 }

```

TODO: describe

```

2358 \def\eq\intertext@env{intertext}
2359 \def\eq\intertext@[#1]{%
2360     \global\def\eq\intertext@opt{#1}%
2361     \ifx\@currenvir\eq\intertext@env
2362         \expandafter\eq\scan@env\expandafter\eq\intertext@inject
2363     \else

```

```

2364 \expandafter\eql@intertext@process
2365 \fi
2366 }

```

TODO: describe

```

2367 \def\eql@intertext@inject{%
2368 \global\edef\eql@intertext@after{%
2369 \noexpand\eql@intertext@process{%
2370 \ifx\eql@scan@body\eql@scan@body@dump
2371 \eql@scan@body@dump
2372 \else
2373 \noexpand\scantokens{\eql@scan@body@dump}%
2374 \fi
2375 }%
2376 }%
2377 }

```

8.4 Line Marks

TODO: describe

```

2378 \def\eql@markline@pos@below{below}
2379 \def\eql@markline@pos@bottom{bottom}
2380 \def\eql@markline@pos@baseline{baseline}
2381 \let\eql@markline@pos\eql@markline@pos@baseline
2382 \let\eql@markline@shift\z@
2383 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2384 \def\eql@markline@symbol{}

```

TODO: describe

```

2385 \def\eql@markline@select#1{%
2386 \let\eql@markline@shift\z@
2387 \eql@setkeys{markline}{#1}%
2388 \eql@markline@print
2389 }

```

TODO: describe

```

2390 \def\eql@markline@print{%
2391 \dimen@=\dimexpr\eql@markline@shift\relax
2392 \ifx\eql@markline@pos\eql@markline@pos@below
2393 \ifdim\dimen@=\z@\else
2394 \penalty\@M
2395 \vskip-\dimen@
2396 \fi
2397 \nointerlineskip
2398 \penalty\@M
2399 \vbox{\hfill\hbox{\eql@markline@symbol}}}%
2400 \else
2401 \ifx\eql@markline@pos\eql@markline@pos@baseline
2402 \advance\dimen@\prevdepth
2403 \fi
2404 \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}}%
2405 \dimen@\prevdepth
2406 \ht\z@\z@
2407 \dp\z@\z@
2408 \nointerlineskip

```

```

2409 \penalty\@M
2410 \vbox{\hfill\box\z@}%
2411 \prevdepth\dimen@
2412 \fi
2413 }

```

TODO: describe

```

2414 \def\eq@markline@inject#1{%
2415 \let\eq@markline@push\eq@false
2416 \ifx\eq@markline@pos\eq@markline@pos@below\else
2417 \ifdefined\eq@tagsleft\else
2418 \ifx\eq@equations@main\eq@multi@main
2419 \ifdefined\eq@numbering@multi
2420 \if@eqnsw
2421 \let\eq@markline@push\eq@true
2422 \fi
2423 \else
2424 \ifnum\eq@row@=\eq@tagpos@row@
2425 \let\eq@markline@push\eq@true
2426 \fi
2427 \fi
2428 \else
2429 \if@eqnsw
2430 \let\eq@markline@push\eq@true
2431 \fi
2432 \fi
2433 \fi
2434 \fi
2435 \ifdefined\eq@markline@push
2436 \global\eq@append\eq@interline@container{%
2437 \eq@append\eq@display@injectbefore{\eq@markline@select{push,#1}}}%
2438 \else
2439 \global\eq@append\eq@interline@container{%
2440 \eq@append\eq@display@injectbefore{\eq@markline@select{#1}}}%
2441 \fi
2442 }

```

TODO: describe

```

2443 \def\eq@markline@amsthm@opt[#1]{\eq@markline@inject{qed,#1}}
2444 \def\eq@markline@amsthm@staropt[#1]{\eq@markline@inject{qed,push,#1}}
2445 \def\eq@markline@amsthm@qed{\eq@teststaropt@tight
2446 \eq@markline@amsthm@staropt\eq@markline@amsthm@opt{}}
2447 \def\eq@markline@amsthm@register#1{\eq@letcs{#1@qed}\eq@markline@amsthm@qed}
2448 \def\eq@markline@amsthm@move#1#2{%
2449 \AddToHook{package/amsthm/after}{%
2450 \eq@letcs{#1@qed\expandafter}\csname#2@qed\endcsname}}

```

9 Column Placement

TODO: describe

9.1 Supporting Definitions

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

respectively:

```
2451 \newcount\eq@shape@pos@
2452 \newdimen\eq@shape@amount@
2453 \let\eq@shape@lastrow\eq@false
```

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

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

```
2454 \newdimen\eq@marginleft@
2455 \newdimen\eq@marginright@
2456 \newdimen\eq@marginleft@min@
2457 \newdimen\eq@centeroffset@
```

9.2 Shape Schemes

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

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

```
2458 \def\eq@shape@tab@default{default}
2459 \def\eq@shape@tab@left{left}
2460 \def\eq@shape@tab@center{center}
2461 \def\eq@shape@tab@right{right}
2462 \def\eq@shape@tab@first{first}
2463 \def\eq@shape@tab@hanging{hanging}
2464 \def\eq@shape@tab@steps{steps}
```

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

```
2465 \let\eq@shape@tab@def\eq@shape@tab@default
2466 \let\eq@shape@tab@\eq@shape@tab@default
2467 \let\eq@shape@tab@l\eq@shape@tab@left
2468 \let\eq@shape@tab@c\eq@shape@tab@center
2469 \let\eq@shape@tab@r\eq@shape@tab@right
2470 \let\eq@shape@tab@rc\eq@shape@tab@first
2471 \let\eq@shape@tab@indent\eq@shape@tab@first
2472 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2473 \let\eq@shape@tab@lcr\eq@shape@tab@hanging
2474 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2475 \let\eq@shape@tab@lcr\eq@shape@tab@steps
```

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

```
2476 \let\eq@shape@mode\eq@shape@tab@default
```

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

```
2477 \def\eq@shape@set#1{%
2478   \ifcsname eq@shape@tab@#1\endcsname
2479     \expandafter\let\expandafter\eq@shape@mode
2480       \csname eq@shape@tab@#1\endcsname
2481   \else
2482     \eq@error{shape '#1' unknown: setting to default}%
2483     \let\eq@shape@mode\eq@shape@tab@default
2484   \fi
2485 }
```


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

```

2486 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
2487 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne@eq@shape@amount@z@}
2488 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2489 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2490 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
2491 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne@eq@shape@amount@z@}
2492 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2493 \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):

```

2494 \def\eq@shape@layoutcenter@first{%
2495   \eq@shape@pos@z@
2496   \eq@shape@amount@z@
2497   \ifnum\eq@totalrows@>\@ne
2498     \ifnum\eq@row@=\@ne
2499       \eq@shape@amount@\eq@indent@
2500     \fi
2501   \fi
2502 }
2503 \def\eq@shape@layoutleft@first{%
2504   \eq@shape@pos@z@
2505   \eq@shape@amount@z@
2506   \ifnum\eq@totalrows@>\@ne
2507     \ifnum\eq@row@=\@ne
2508       \eq@shape@amount@\eq@indent@
2509     \fi
2510   \fi
2511 }

```

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:

```

2512 \def\eq@shape@layoutcenter@hanging{%
2513   \eq@shape@pos@z@
2514   \eq@shape@amount@\eq@indent@
2515   \ifnum\eq@totalrows@>\@ne
2516     \ifnum\eq@row@=\@ne
2517       \eq@shape@amount@z@
2518     \fi
2519   \fi
2520 }
2521 \def\eq@shape@layoutleft@hanging{%
2522   \eq@shape@pos@z@
2523   \eq@shape@amount@z@
2524   \ifnum\eq@totalrows@>\@ne
2525     \ifnum\eq@row@=\@ne
2526       \eq@shape@amount@-\eq@indent@
2527     \fi
2528   \fi
2529 }

```

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:

```

2530 \def\eq@shape@layoutcenter@steps{%
2531   \eq@shape@amount@z@
2532   \eq@shape@pos@\@ne
2533   \ifnum\eq@totalrows@>\@ne
2534     \ifnum\eq@row@=\@ne
2535       \eq@shape@pos@z@
2536     \fi
2537     \ifnum\eq@row@=\eq@totalrows@
2538       \eq@shape@pos@tw@
2539     \fi
2540   \fi
2541 }
2542 \def\eq@shape@layoutleft@steps{%
2543   \eq@shape@pos@z@
2544   \eq@shape@amount@z@
2545   \ifnum\eq@totalrows@>\@ne
2546     \ifnum\eq@row@=\@ne
2547       \eq@shape@amount@-\eq@indent@
2548     \fi
2549     \ifnum\eq@row@=\eq@totalrows@
2550       \eq@shape@amount@\eq@indent@
2551     \fi
2552   \fi
2553 }
```

`\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`:

```

2554 \let\eq@shape@eval\undefined
2555 \def\eq@shape@select{%
2556   \expandafter\let\expandafter\eq@shape@eval
2557     \csname eq@shape%
2558       @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2559       @\eq@shape@mode\endcsname
2560 }
```

`\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`

```

2561 \protected\def\eq@shape@alignleft{%
2562   \global\eq@append\eq@cell@container{\eq@shape@pos@z@}%
2563   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2564 \protected\def\eq@shape@aligncenter{%
2565   \global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}%
2566   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2567 \protected\def\eq@shape@alignright{%
2568   \global\eq@append\eq@cell@container{\eq@shape@pos@tw@}%
2569   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2570 \def\eq@shape@align@testpar#1{%
2571   \eq@ifstar@tight{#1[\eq@indent@]}%
2572   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2573   {\eq@testopt@tight{#1}\z@}}%
2574 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}
```

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

```

2575 \protected\def\eq@shape@alignamount{%
```

```

2576 \eql@ampprotecttwo\eql@ifstar@tight
2577 \eql@shape@alignamount@set\eql@shape@alignamount@add}
2578 \def\eql@shape@alignamount@add#1{%
2579 \global\eql@appendexpand\eql@cell@container{%
2580 \advance\eql@shape@amount@the\glueexpr#1\relax\relax}}
2581 \def\eql@shape@alignamount@set#1{%
2582 \global\eql@appendexpand\eql@cell@container{%
2583 \eql@shape@amount@the\glueexpr#1\relax\relax}}
2584 \def\eql@shape@align@enable{%
2585 \let\shoveleft\eql@shape@alignleft
2586 \let\shovecenter\eql@shape@aligncenter
2587 \let\shoveright\eql@shape@alignright
2588 \let\shoveby\eql@shape@alignamount
2589 }

```

TODO: describe

```

2590 \protected\def\eql@shape@align@default{%
2591 \eql@warn@here{\shove...}%
2592 \eql@ampprotect\eql@shape@align@testpar\eql@gobbleopt}
2593 \protected\def\eql@shape@alignamount@default{%
2594 \eql@warn@here{\shove...}%
2595 \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble}
2596 \def\eql@shape@align@disable{%
2597 \let\shoveleft\eql@shape@align@default
2598 \let\shovecenter\eql@shape@align@default
2599 \let\shoveright\eql@shape@align@default
2600 \let\shoveby\eql@shape@alignamount@default
2601 }

```

9.3 Width Data

width@block@ (*dimen*)

```

2602 \newdimen\eql@tagwidth@block@
2603 \newdimen\eql@tagheight@block@
2604 \newdimen\eql@tagdepth@block@

```

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

```

2605 \let\eql@dimensions@tab\@empty

```

$\text{eql@dimensions@reset}$

```

2606 \def\eql@dimensions@reset{%
2607 \let\eql@dimensions@tab\@empty
2608 \eql@tagwidth@max@z@
2609 \eql@tagrows@z@
2610 }

```

$\text{eql@dimensions@add}$

```

2611 \def\eql@dimensions@add#1{%
2612 \global\eql@appendexpand\eql@dimensions@tab{#1}%
2613 }

```

$\text{eql@dimensions@addreg}$

```

2614 \def\eql@dimensions@addreg#1{#1\the#1\relax}

```

@dimensions@startrow

```
2615 \def\eqldimensions@startrow{%
2616   \eqldimensions@add{\eqldimensions@addreg\eqldrow@}%
2617 }
```

@dimensions@savecell

```
2618 \def\eqldimensions@savecell{%
2619   \eqldimensions@add{%
2620     \eqldimensions@addreg\eqldshape@pos@
2621     \eqldimensions@addreg\eqldcellwidth@
2622     \eqldimensions@addreg\eqldshape@amount@
2623     \noexpand\eqldimensions@cellcall
2624   }%
2625 }
```

@dimensions@savesep

```
2626 \def\eqldimensions@savesep{%
2627   \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2628 }
```

@dimensions@endrow

```
2629 \def\eqldimensions@endrow{%
2630   \eqldimensions@add{,%
2631     \eqldimensions@addreg\eqldtagwidth@
2632     \eqldimensions@addreg\eqldline@height@
2633     \eqldimensions@addreg\eqldline@depth@
2634     \eqldimensions@addreg\eqldline@interline@
2635   ;}%
2636 }
```

@dimensions@saveblocktag

```
2637 \def\eqldimensions@saveblocktag{%
2638   \eqldimensions@add{\eqldrow@0\relax,%
2639     \eqldtagwidth@block@\the\eqldtagwidth@\relax
2640     \eqldtagheight@block@\the\ht\eqldtagbox@\relax
2641     \eqldtagdepth@block@\the\dp\eqldtagbox@\relax
2642     \eqldimensions@addreg\eqldtagpos@shift@
2643     \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
2644     \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
2645   ;}%
2646   \global\eqldtagwidth@max@\eqldtagwidth@
2647   \global\eqldtaggrows@\@ne
2648 }
```

@dimensions@savenoblocktag

```
2649 \def\eqldimensions@savenoblocktag{%
2650   \eqldimensions@add{\eqldrow@0\relax,;%
2651 }
```

@eqldimensions@for

```
2652 \def\eqldimensions@for#1{%
2653   \def\eqldimensions@forcall{#1}%
2654   \expandafter\eqldimensions@forstep\eqldimensions@tab
2655 }
```

l@dimensions@forstep

```

2656 \def\eq@dimensions@forstep\eq@row@#1\relax#2,##3;%
2657 \eq@row@#1\relax
2658 \ifnum\eq@row@=\z@\else
2659   #3%
2660   \def\eq@dimensions@cells{#2}%
2661   \eq@dimensions@forall
2662   \expandafter\eq@dimensions@forstep
2663 \fi
2664 }

```

\eq@dimensions@get

```

2665 \def\eq@dimensions@get#1{%
2666 \eq@row@#1\relax
2667 \expandafter\eq@dimensions@getdef\expandafter{\the\eq@row@}%
2668 \expandafter\eq@dimensions@getparse\eq@dimensions@tab\@nil
2669 }

```

ql@dimensions@getdef

```

2670 \def\eq@dimensions@getdef#1{%
2671 \def\eq@dimensions@getparse
2672   ##1\eq@row@#1\relax##2,##3;##4\@nil{%
2673   ##3%
2674   \def\eq@dimensions@cells{##2}%
2675   }%
2676 }

```

\eq@colwidth@tab

```

2677 \let\eq@colwidth@tab\@empty

```

\eq@colwidth@get

```

2678 \def\eq@colwidth@get#1{%
2679 \ifcase\expandafter#1\eq@colwidth@tab\else\z@\fi
2680 }

```

\eq@colwidth@save

```

2681 \def\eq@colwidth@save#1{%
2682 \edef\eq@colwidth@tab{%
2683 \noexpand\or\the#1%
2684 \unexpanded\expandafter{\eq@colwidth@tab}%
2685 }%
2686 }

```

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

```

2687 \def\eq@dimensions@calc{%
2688 \eq@column@\z@
2689 \eq@line@pos@\z@

```

```

2690 \eql@line@possep@\z@
2691 \eql@line@avail@\eql@totalwidth@
2692 \eql@line@availsep@\eql@intercolumns@
2693 \eql@line@width@\z@
2694 \eql@line@widthsep@\z@
2695 \let\eql@dimensions@cellcall\eql@dimensions@calc@call
2696 \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
2697 \eql@dimensions@cells
2698 }

```

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

```

2699 \def\eql@dimensions@calc@callsep{%
2700   \advance\eql@line@possep@\@ne
2701 }%

```

`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?!)

```

2702 \def\eql@dimensions@calc@call{%
2703   \advance\eql@column@\@ne
2704   \ifnum\eql@totalcolumns@=\@ne
2705     \dimen@\eql@totalwidth@
2706   \else
2707     \dimen@\eql@colwidth@get\eql@column@\relax
2708   \fi
2709   \ifdim\eql@cellwidth@>\z@
2710     \ifdim\eql@line@width@=\z@
2711       \eql@line@avail@\eql@line@pos@
2712       \eql@line@availsep@\eql@line@possep@
2713       \ifcase\eql@shape@pos@
2714       \or
2715         \advance\eql@line@avail@\dimexpr
2716           (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2717       \or
2718         \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
2719       \fi
2720       \advance\eql@line@avail@\eql@shape@amount@
2721     \fi
2722     \eql@line@width@\eql@line@pos@
2723     \eql@line@widthsep@\eql@line@possep@
2724     \ifcase\eql@shape@pos@
2725     \advance\eql@line@width@\eql@cellwidth@
2726     \or
2727     \advance\eql@line@width@\dimexpr
2728       (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2729     \or
2730     \advance\eql@line@width@\dimen@
2731     \fi
2732     \advance\eql@line@width@\eql@shape@amount@
2733   \fi
2734   \advance\eql@line@pos@\dimen@
2735 }

```

9.4 Best Line Selection

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

```
2736 \let\eql@numbering@best@auto\eql@false
```

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

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

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

```
2737 \newcount\eql@numbering@best@row@
```

```
2738 \newdimen\eql@numbering@best@space@
```

```
2739 \let\eql@numbering@best@use\eql@false
```

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

```
2740 \def\eql@numbering@best@find{%
2741   \eql@numbering@best@row@ \z@
2742   \eql@numbering@best@space@ \z@
2743   \eql@dimensions@for{%
2744     \eql@dimensions@calc
2745     \ifdefined\eql@tagsleft
2746       \dimen@ \eql@line@avail@
2747     \else
2748       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
2749     \fi
2750     \ifdim \dimen@ > \eql@numbering@best@space@
2751       \eql@numbering@best@row@ \eql@row@
2752       \eql@numbering@best@space@ \dimen@
2753     \fi
2754   }%
2755   \ifnum \eql@numbering@best@row@ > \z@
2756     \eql@tagpos@row@ \eql@numbering@best@row@
2757     \let\eql@tagpos@continuous\eql@false
2758     \eql@tagpos@prevrow@ \z@
2759   \fi
2760 }
```

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

```
2761 \def\eql@numbering@best@test#1{%
2762   \eql@dimensions@get#1%
2763   \eql@dimensions@calc
2764   \ifdefined\eql@tagsleft
2765     \dimen@ \dimexpr \eql@line@avail@
2766       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
2767   \else
2768     \dimen@ \dimexpr \displaywidth@ - \eql@line@width@
2769       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
2770   \fi
2771   \ifdim \dimen@ < \eql@tagwidth@block@
2772     \let\eql@numbering@best@use\eql@true
2773   \fi
2774 }
```

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

```
2775 \def\eql@numbering@best@eval{%
2776   \ifdefined\eql@numbering@best@auto
2777     \ifdefined\eql@numbering@best@use\else
2778       \ifdefined\eql@numbering@multi\else
```

```

2779         \ifnum\eql@tagpos@row@>\z@
2780             \eql@numbering@best@test\eql@tagpos@row@
2781         \fi
2782         \ifnum\eql@tagpos@prevrow@>\z@
2783             \eql@numbering@best@test\eql@tagpos@prevrow@
2784         \fi
2785     \fi
2786 \fi
2787 \fi
2788 \ifdefined\eql@numbering@best@use
2789     \eql@numbering@best@find
2790 \fi
2791 }

```

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

```

2792 \def\eql@adjust@calc@tagmargin{%
2793     \ifdefined\eql@tagmargin@val
2794         \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2795     \else
2796         \eql@tagmargin@\eql@tagwidth@max@
2797         \ifdim\eql@tagmargin@>\z@
2798             \advance\eql@tagmargin@-\eql@tagsepmin@
2799         \fi
2800     \fi

2801     \dimen@\eql@tagrows@\p@
2802     \ifnum\eql@totalrows@=\@ne
2803         \ifnum\eql@tagrows@=\@ne
2804             \advance\dimen@1sp\relax
2805         \fi
2806     \fi
2807     \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
2808         \eql@tagmargin@\z@
2809     \fi

2810     \@tempdima\dimexpr\displaywidth
2811         -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
2812     \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
2813     \ifdim\@tempdimb>\z@
2814         \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
2815             \eql@tagmargin@\z@
2816         \fi
2817     \fi
2818 }

```

9.6 Single Column

ql@adjust@calc@lines

```

2819 \def\eql@adjust@calc@lines{%
2820     \eql@totalcolumns@\@ne

```



```

2821 \eql@intercolumns@\z@
2822 \eql@colsep@\z@
2823 \ifdefined\eql@layoutleft
2824 \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2825 \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2826 \ifdim\eql@marginleft@<\eql@marginleft@min@
2827 \eql@marginleft@\eql@marginleft@min@
2828 \fi
2829 \dimen@\glueexpr\eql@layoutleftmarginmax\relax
2830 \ifdim\eql@marginleft@>\dimen@
2831 \eql@marginleft@\dimen@
2832 \fi
2833 \eql@marginright@\z@
2834 \eql@centeroffset@\z@
2835 \else
2836 \eql@adjust@calc@tagmargin
2837 \ifdefined\eql@paddingleft@val
2838 \eql@marginleft@\dimexpr
2839 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2840 -\glueexpr\eql@paddingleft@val\relax\relax
2841 \ifdim\eql@marginleft@<\z@
2842 \eql@marginleft@\z@
2843 \fi
2844 \else
2845 \eql@marginleft@\z@
2846 \fi
2847 \ifdefined\eql@paddingright@val
2848 \eql@marginright@\dimexpr
2849 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2850 -\glueexpr\eql@paddingright@val\relax\relax
2851 \ifdim\eql@marginright@<\z@
2852 \eql@marginright@\z@
2853 \fi
2854 \else
2855 \eql@marginright@\z@
2856 \fi
2857 \ifdim\eql@tagmargin@>\z@
2858 \ifdefined\eql@tagsleft
2859 \ifdim\eql@marginleft@<\eql@tagsepmin@
2860 \eql@marginleft@\eql@tagsepmin@
2861 \fi
2862 \advance\eql@marginleft@\eql@tagmargin@
2863 \advance\eql@centeroffset@\eql@tagmargin@
2864 \else
2865 \ifdim\eql@marginright@<\eql@tagsepmin@
2866 \eql@marginright@\eql@tagsepmin@
2867 \fi
2868 \advance\eql@marginright@\eql@tagmargin@
2869 \advance\eql@centeroffset@-\eql@tagmargin@
2870 \fi
2871 \fi
2872 \eql@marginleft@min@\z@
2873 \eql@centeroffset@\dimexpr\eql@marginright@-\eql@marginleft@
2874 \ifdefined\eql@tagsleft+ \else -\fi \eql@tagmargin@ \relax
2875 \fi

2876 \eql@totalwidth@\dimexpr\displaywidth
2877 -\eql@marginleft@-\eql@marginright@ \relax
2878 }

```

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:

```
2879 \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:

```
2880 \ifnum\eql@totalcolumns@<\thr@@
2881 \eql@totalcolumns@tw@
2882 \let\eql@columns@fulllength\eql@false
2883 \fi
```

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

```
2884 \eql@intercolumns@ \numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2885 \eql@colsepmin@ \glueexpr\eql@colsepmin@val\relax
```

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

```
2886 \ifdefined\eql@layoutleft
2887 \eql@marginleft@ \glueexpr\eql@layoutleftmargin\relax
2888 \eql@marginleft@min@ \glueexpr\eql@layoutleftmarginmin\relax
2889 \ifdim\eql@marginleft@<\eql@marginleft@min@
2890 \eql@marginleft@ \eql@marginleft@min@
2891 \fi
2892 \else
```

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

```
2893 \eql@adjust@calc@tagmargin
2894 \ifdefined\eql@columns@fulllength
2895 \ifdim\eql@tagmargin@>\z@
2896 \advance\eql@tagmargin@ \eql@tagsepmin@
2897 \fi
2898 \fi
2899 \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2900 -\eql@intercolumns@ \eql@colsepmin@ \relax
2901 \eql@tagmargin@ \z@
2902 \fi
2903 \eql@marginleft@min@ \z@
2904 \fi
```

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

```
2905 \ifnum\eql@intercolumns@>\z@
```

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

```

2906 \eq@colsep@\dimexpr\displaywidth-\eq@totalwidth@\relax
2907 \ifdefined\eq@layoutleft
2908 \advance\eq@colsep@-\eq@marginleft@
2909 \else
2910 \advance\eq@colsep@-\eq@tagmargin@
2911 \fi
2912 \count@\eq@intercolumns@
2913 \ifdefined\eq@columns@fulllength\else
2914 \ifdefined\eq@layoutleft
2915 \advance\count@\@ne
2916 \else
2917 \advance\count@\tw@
2918 \fi
2919 \fi
2920 \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:

```

2921 \ifdim\eq@colsep@<\eq@colsepmin@
2922 \eq@colsep@\eq@colsepmin@
2923 \else
2924 \ifdefined\eq@columns@fulllength\else
2925 \dimen@\glueexpr\eq@colsepmax@val\relax
2926 \ifdim\eq@colsep@>\dimen@
2927 \eq@colsep@\dimen@
2928 \fi
2929 \fi
2930 \fi
2931 \else

```

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

```

2932 \eq@colsep@\eq@colsepmin@
2933 \fi

```

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

```

2934 \ifdefined\eq@layoutleft

```

Set the default value:

```

2935 \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:

```

2936 \dimen@\dimexpr\displaywidth-\eq@totalwidth@
2937 -\eq@intercolumns@\eq@colsep@\relax
2938 \ifdim\dimen@<\eq@marginleft@
2939 \ifdim\dimen@<\eq@marginleft@min@
2940 \eq@marginleft@\eq@marginleft@min@
2941 \else
2942 \eq@marginleft@\dimen@
2943 \fi
2944 \fi
2945 \fi
2946 \else

```

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

```

2947 \ifdefined\eql@columns@fulllength
2948 \eql@marginleft@<z@

```

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

```

2949 \else
2950 \eql@marginleft@<\dimexpr(\displaywidth-\eql@totalwidth@
2951 -\eql@intercolumns@<\eql@colsep@-\eql@tagmargin@)/\tw@<\relax
2952 \ifdim\eql@marginleft@<<z@
2953 \eql@marginleft@<z@
2954 \fi
2955 \fi

```

Add tag margin in case of left tags:

```

2956 \ifdefined\eql@tagsleft
2957 \advance\eql@marginleft@<\eql@tagmargin@
2958 \fi
2959 \fi

```

Find the best row for tag placement:

```

2960 \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:

```

2961 \ifdefined\eql@tagsleft
2962 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
2963 \else
2964 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2965 \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

```

2966 \ifdefined\eql@numbering@multi
2967 \eql@dimensions@for{%
2968 \ifdim\eql@tagwidth@>z@
2969 \eql@dimensions@calc
2970 \eql@adjust@columns@test
2971 \fi
2972 }%
2973 \else
2974 \ifnum\eql@tagpos@row@>z@
2975 \ifnum\eql@tagpos@row@>\eql@totalrows@\else
2976 \eql@dimensions@get\eql@tagpos@row@
2977 \eql@tagwidth@\eql@tagwidth@block@
2978 \eql@dimensions@calc
2979 \eql@adjust@columns@test
2980 \fi
2981 \fi
2982 \ifnum\eql@tagpos@prevrow@>z@
2983 \eql@dimensions@get\eql@tagpos@prevrow@
2984 \eql@tagwidth@\eql@tagwidth@block@
2985 \eql@dimensions@calc
2986 \eql@adjust@columns@test
2987 \fi

```

```
2988 \fi
```

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

```
2989 \advance\eq@totalwidth@\dimexpr
2990 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
2991 }
```

Placement for Right Tags.

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

```
2992 \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?

```
2993 \@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.

```
2994 \ifdim\@tempdima<\dimexpr
2995 \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).

```
2996 \ifdim\@tempdima<\dimexpr
2997 \eq@marginleft@min@+\eq@line@widthsep@\eq@colsepmin@\relax\else
```

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

```
2998 \eq@adjust@columns@modify@tagsright
2999 \fi
3000 \fi
3001 }
```

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

```
3002 \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.

```
3003 \ifnum\eq@line@widthsep@>\z@
3004 \dimen@\@tempdima
3005 \count@\eq@line@widthsep@
3006 \ifdefined\eq@layoutleft
3007 \advance\dimen@-\eq@marginleft@
3008 \else
3009 \ifdefined\eq@columns@fulllength\else
3010 \advance\count@\@ne
3011 \fi
3012 \fi
3013 \divide\dimen@\count@
```

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

```

3014 \ifdim\dimen@<\eq\colsep@
3015 \ifdim\dimen@<\eq\colsepmin@
3016 \eq\colsep@\eq\colsepmin@
3017 \else
3018 \eq\colsep@\dimen@
3019 \fi
3020 \fi
3021 \fi

```

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

```

3022 \dimen@\dimexpr\@tempdima-\eq\line@widthsep@\eq\colsep@\relax
3023 \ifdim\eq\marginleft@>\dimen@
3024 \eq\marginleft@\dimen@
3025 \fi
3026 }

```

Placement for Left Tags.

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

```

3027 \def\eq\adjust@columns@test@tagsleft{%

```

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

```

3028 \count@\numexpr\eq\intercolumns@-\eq\line@availsep@\relax
3029 \@tempdima\dimexpr\eq>tagwidth@-\eq\line@avail@\relax

```

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

```

3030 \ifdim\@tempdima>\dimexpr
3031 \eq\marginleft@+\eq\line@availsep@\eq\colsep@\relax

```

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

```

3032 \ifdim\eq>tagwidth@<%
3033 \ifdefined\eq\layoutleft
3034 \glueexpr\eq\layoutleftmarginmax\relax
3035 \else
3036 \displaywidth
3037 \fi

```

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

```

3038 \ifdim\@tempdima>\dimexpr
3039 \displaywidth-\eq\totalwidth@-\count@\eq\colsepmin@\relax\else

```

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

```

3040 \eq\adjust@columns@modify@tagsleft
3041 \fi
3042 \fi
3043 \fi
3044 }

```

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

```
3045 \def\eql@adjust@columns@modify@tagsleft{%
```

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

```
3046 \ifnum\count@>\z@
3047 \dimen@ \dimexpr\displaywidth-\eql@totalwidth@-\@tempdima\relax
3048 \ifdefined\eql@columns@fulllength\else
3049 \advance\count@\@ne
3050 \fi
3051 \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.

```
3052 \ifdim\dimen@<\eql@colsep@
3053 \ifdim\dimen@<\eql@colsepmin@
3054 \dimen@\eql@colsepmin@
3055 \fi
3056 \advance\dimen@-\eql@colsep@
3057 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3058 \advance\eql@colsep@\dimen@
3059 \fi
3060 \fi
```

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

```
3061 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3062 \ifdim\eql@marginleft@<\dimen@
3063 \eql@marginleft@\dimen@
3064 \fi
3065 }
```

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:

```
3066 \ifdefined\inf@bad\else%
3067 \newcount\inf@bad
3068 \inf@bad1000000\relax
3069 \fi
```

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

```
3070 \let\eql@restore@hfuzz\@empty
3071 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

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

```
3072 \newcount\eqalignbadness@
3073 \newcount\eqtagbadness@
3074 \newcount\eqarrange@badness@
3075 \eqalignbadness@\inf@bad
3076 \eqtagbadness@\inf@bad
```

10.2 Arrangement Methods

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

```
3077 \def\eqarrange@try#1{%
3078   \ifdim#1>\dimexpr\displaywidth-\eqcellwidth@\relax
3079     \setbox\eqcellbox@\hbox to\displaywidth{%
3080       \unhbox\eqcellbox@\unkern\kern#1}%
3081     \eqarrange@badness@badness
3082   \else
3083     \eqarrange@badness@m@ne
3084   \fi
3085 }
```

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

TODO: adjust

```
3086 \def\eqarrange@print#1#2{%
3087   \eq@restore@hfuzz
3088   \if@eqnsw
3089     \ifdefined\eq@tagsleft
3090       \eq@tagbox@print@tagsleft
3091     \fi
3092   \fi
3093   \hbox to\displaywidth{%
3094     #1%
3095     \unhbox\eqcellbox@\unkern
3096     #2%
3097     \eq@tagging@mathaddlast
3098   }%
3099   \if@eqnsw
3100     \ifdefined\eq@tagsleft\else
3101       \eq@tagbox@print@tagsright
3102     \fi
3103   \fi
3104 }
```

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

```
3105 \def\eqarrange@print@alignleft#1#2{%
3106   \eqdisplay@firstavail@set{\dimexpr#1\relax}%
3107   \eqarrange@print{\kern#1}{\kern#2}%
3108 }
```



```

3109 \def\eql@arrange@print@alignright#1#2{%
3110   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3111   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3112 }

3113 \def\eql@arrange@print@aligncenter#1{%
3114   \eql@display@firstavail@set{\dimexpr
3115     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3116   \ifdim#1>\z@
3117     \eql@arrange@print{\kern#1\hfil}{}%
3118   \else
3119     \eql@arrange@print{\hfil}{\kern-#1}%
3120   \fi
3121 }

```

`\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:

```

3122 \def\eql@arrange@init{%
3123   \eql@save@hfuzz
3124   \hfuzz\maxdimen
3125   \eql@shape@select
3126 }

```

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

```

3127 \def\eql@arrange@print@line{%
3128   \eql@tagging@tagaddbox
3129   \csname eql@arrange%
3130     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3131     @init\endcsname
3132   \csname eql@arrange%
3133     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3134     @\ifdefined\eql@tagpos@reserve
3135       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3136       notag\fi\endcsname
3137 }

```

10.3 Central Alignment

TODO: describe

```

3138 \def\eql@arrange@aligncenter@init{%
3139   \eql@tagging@aligncenter
3140   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3141     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3142 }

```

TODO: describe

```

3143 \def\eql@arrange@aligncenter@notag{%
3144   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3145     \ifdim\eql@line@offset@<\eql@marginleft@min@
3146       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3147     \else
3148       \eql@line@offset@
3149     \fi

```

```

3150 \eql@arrange@print@aligncenter\eql@line@offset@
3151 \else
3152 \ifdim\eql@line@offset@<\eql@marginleft@min@
3153 \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3154 \else
3155 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3156 \fi
3157 \fi
3158 }

```

TODO: describe

```

3159 \def\eql@arrange@aligncenter@tagsright{%
3160 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3161 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3162 \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3163 \else
3164 \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
3165 \fi
3166 \eql@arrange@print@aligncenter\eql@line@offset@
3167 \else
3168 \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
3169 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3170 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3171 \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3172 \else
3173 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3174 \fi
3175 \else
3176 \let\eql@tagpos@reserve\eql@false
3177 \eql@arrange@aligncenter@notag
3178 \fi
3179 \fi
3180 }

```

```

3181 \def\eql@arrange@aligncenter@tagsleft{%
3182 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3183 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3184 \ifdim\eql@line@offset@<\eql@tagwidth@
3185 \dimexpr\tw@\eql@tagwidth@-\eql@line@offset@\relax
3186 \else
3187 \eql@line@offset@
3188 \fi
3189 \eql@arrange@print@aligncenter\eql@line@offset@
3190 \else
3191 \eql@arrange@try\eql@tagwidth@
3192 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3193 \ifdim\eql@line@offset@<\eql@tagwidth@
3194 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3195 \else
3196 \eql@arrange@print@alignright\eql@tagwidth@\z@
3197 \fi
3198 \else
3199 \let\eql@tagpos@reserve\eql@false
3200 \eql@arrange@aligncenter@notag
3201 \fi
3202 \fi
3203 \else
3204 \eql@arrange@aligncenter@notag

```

```

3205 \fi
3206 }

```

10.4 Left Alignment

```

3207 \def\eql@arrange@alignleft@init{%
3208   \eql@tagging@alignleft
3209   \eql@line@offset@dimexpr\eql@marginleft@+\eql@shape@amount@\relax
3210   \ifdim\eql@line@offset@<\eql@marginleft@min@
3211     \eql@line@offset@\eql@marginleft@min@
3212   \fi
3213 }

3214 \def\eql@arrange@alignleft@notag{%
3215   \ifdim\eql@line@offset@>\eql@marginleft@min@
3216     \eql@arrange@try\eql@line@offset@
3217     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3218       \eql@arrange@print@alignleft\eql@line@offset@\z@
3219     \else
3220       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3221     \fi
3222   \else
3223     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3224   \fi
3225 }

3226 \def\eql@arrange@alignleft@tagsright{%
3227   \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3228   \ifnum\eql@arrange@badness@<\eql@alignbadness@
3229     \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3230   \else
3231     \ifdim\eql@line@offset@>\eql@marginleft@min@
3232       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3233     \fi
3234     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3235       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3236     \else
3237       \let\eql@tagpos@reserve\eql@false
3238       \eql@arrange@alignleft@notag
3239     \fi
3240   \fi
3241 }

3242 \def\eql@arrange@alignleft@tagsleft{%
3243   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3244     \ifdim\eql@line@offset@>\eql@tagwidth@
3245       \eql@arrange@try\eql@line@offset@
3246       \ifnum\eql@arrange@badness@<\eql@alignbadness@
3247         \eql@arrange@print@alignleft\eql@line@offset@\z@
3248       \else
3249         \eql@arrange@try\eql@tagwidth@
3250       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3251         \eql@arrange@print@alignright\eql@tagwidth@\z@
3252       \else
3253         \let\eql@tagpos@reserve\eql@false
3254         \eql@arrange@print@alignright\eql@marginleft@min@\z@
3255       \fi
3256     \fi
3257   \else
3258     \eql@arrange@try\eql@tagwidth@

```

```

3259     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3260     \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3261   \else
3262     \let\eqL@tagpos@reserve\eqL@false
3263     \eqL@arrange@alignleft@notag
3264   \fi
3265 \fi
3266 \else
3267   \eqL@arrange@alignleft@notag
3268 \fi
3269 }

```

10.5 Right Alignment

```

3270 \def\eqL@arrange@alignright@init{%
3271   \eqL@tagging@alignright
3272   \eqL@line@offset@dimexpr\eqL@marginright@-\eqL@shape@amount@relax
3273   \ifdim\eqL@line@offset@<z@
3274     \eqL@line@offset@z@
3275   \fi
3276 }

```

TODO: describe

```

3277 \def\eqL@arrange@alignright@notag{%
3278   \ifdim\eqL@line@offset@>z@
3279     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3280     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3281       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3282     \else
3283       \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3284     \fi
3285   \else
3286     \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3287   \fi
3288 }

```

TODO: describe

```

3289 \def\eqL@arrange@alignright@tagsright{%
3290   \ifdim\eqL@line@offset@>\eqL@tagwidth@
3291     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3292     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3293       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3294     \else
3295       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3296       \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3297         \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
3298       \else
3299         \let\eqL@tagpos@reserve\eqL@false
3300         \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3301       \fi
3302     \fi
3303   \else
3304     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3305     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3306       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3307     \else
3308       \let\eqL@tagpos@reserve\eqL@false
3309       \eqL@arrange@alignright@notag
3310     \fi

```

```

3311 \fi
3312 }

```

TODO: describe

```

3313 \def\eqL@arrange@alignright@tagsleft{%
3314 \ifdim\eqL@tagwidth@>\eqL@marginleft@min@
3315 \eqL@arrange@try{\dimexpr\eqL@line@offset@+\eqL@tagwidth@\relax}%
3316 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3317 \eqL@arrange@print@alignright\eqL@tagwidth@\eqL@line@offset@
3318 \else
3319 \ifdim\eqL@line@offset@>\z@
3320 \eqL@arrange@try\eqL@tagwidth@
3321 \fi
3322 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3323 \eqL@arrange@print@alignleft\eqL@tagwidth@\z@
3324 \else
3325 \let\eqL@tagpos@reserve\eqL@false
3326 \eqL@arrange@alignright@notag
3327 \fi
3328 \fi
3329 \else
3330 \eqL@arrange@alignright@notag
3331 \fi
3332 }

```

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

`\eqL@box@cr`

```

3333 \protected\def\eqL@box@cr{\eqL@srbgroup%
3334 \eqL@ifnextgobble@tight~%
3335 {\let\eqL@punct@line@empty\eqL@box@cr@opt}%
3336 \eqL@box@cr@opt}
3337 \def\eqL@box@cr@opt{\eqL@ifnextchar@tight[\eqL@box@cr@skip\eqL@box@cr@plain}
3338 \def\eqL@box@cr@plain{\eqL@sregroup\eqL@box@cr@}
3339 \def\eqL@box@cr@skip[#1]{\eqL@sregroup
3340 \eqL@box@cr@
3341 \noalign{%
3342 \vskip\glueexpr#1\relax
3343 }%
3344 }
3345 \def\eqL@box@cr@{%
3346 \eqL@punct@apply@line
3347 \eqL@hook@lineout
3348 \eqL@box@lastcell
3349 \cr
3350 }

```

11.2 Stacked Mode

```
3351 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}
```

TODO: templates

```
3352 \def\eql@box@open@stacked{%
3353   \eql@shape@align@enable
3354   \let\eql@box@lastcell\eql@box@lastcell@stacked
3355   \everycr{\noalign{%
3356     \eql@verbose@info\eql@verbose@msg@startline
3357     \global\advance\eql@row@ \@ne
3358   }}%
3359   \tabskip\z@skip
3360   \halign\bgroup
3361     &%
3362     \global\let\eql@cell@container\@empty
3363     \setbox\eql@cellbox@\hbox{%
3364       \eql@strut@cell
3365       \@lign
3366       $\m@th\displaystyle
3367       \eql@hook@colin
3368       ##%
3369       \eql@punct@apply@col
3370       \eql@hook@colout
3371       \eql@tagging@mathsave
3372       $%
3373       \eql@tagging@mathaddlast
3374     }%
3375     \ifdefined\eql@shape@lastrow
3376       \eql@totalrows@\eql@row@
3377     \fi
3378     \eql@shape@eval
3379     \eql@cell@container
3380     \ifdefined\eql@frame@cmd
3381       \ifcase\eql@shape@pos@
3382         \eql@frame@measure
3383         \advance\eql@shape@amount@-\eql@frame@margin@
3384       \or\or
3385         \eql@frame@measure
3386         \advance\eql@shape@amount@+\eql@frame@margin@
3387       \fi
3388       \eql@frame@print
3389     \fi
3390     \ifcase\eql@shape@pos@
3391       \kern\eql@shape@amount@
3392       \box\eql@cellbox@
3393       \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3394       -\eql@shape@amount@+\@flushglue\relax
3395       \eql@tagging@alignleft
3396     \or
3397       \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3398       \box\eql@cellbox@
3399       \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3400       \eql@tagging@aligncenter
3401     \or
3402       \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3403       +\eql@shape@amount@+\@flushglue\relax
3404       \box\eql@cellbox@
3405       \kern-\eql@shape@amount@
```

```

3406         \eql@tagging@alignright
3407     \fi
3408     \tabskip\eql@colsep@\relax
3409 \crrc
3410 \noalign{%
3411     \global\let\eql@shape@lastrow\eql@false
3412     \eql@hook@blockbefore
3413 }%
3414 \eql@hook@blockin
3415 }
3416 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

11.3 Aligned Mode

```

3417 \def\eql@box@lastcell@odd{%
3418     &\omit
3419     \eql@prevwidth@\wd\eql@cellbox@
3420     \let\eql@frame@cmd\eql@frame@prevcmd
3421     \ifdefined\eql@frame@cmd
3422         \eql@frame@measure
3423         \advance\eql@prevwidth@\eql@frame@margin@
3424         \eql@frame@print
3425     \fi
3426     \kern-\eql@prevwidth@
3427     \unhbox\eql@cellbox@
3428     \hfil
3429     &\omit\kern-\eql@colsep@
3430 }%
3431 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}
3432 \def\eql@verbose@msg@startline@aligned{starting new line}
3433 \def\eql@box@open@aligned{%
3434 % \TODO templates
3435     \eql@shape@align@disable
3436     \let\eql@box@lastcell@empty
3437     \everycr{\noalign{%
3438         \eql@verbose@info\eql@verbose@msg@startline@aligned
3439     }}%
3440     \tabskip\z@skip
3441     \halign\bgroup
3442         &%
3443         \let\eql@box@lastcell\eql@box@lastcell@odd
3444         \global\let\eql@cell@container@empty
3445         \global\setbox\eql@cellbox@\hbox{%
3446             \eql@strut@cell
3447             \@lign
3448             $\m@th\displaystyle
3449             \eql@hook@colin
3450             ##%
3451             \eql@class@innerleft
3452             \eql@hook@innerleft
3453             \eql@tagging@mathsave
3454             $%
3455             \eql@tagging@mathaddlast
3456         }%
3457         \eql@cell@container
3458         \hfil
3459         \kern\wd\eql@cellbox@
3460         \ifdefined\eql@frame@cmd

```

```

3461      \eql@frame@measure
3462      \kern\eql@frame@margin@
3463      \fi
3464      \global\let\eql@frame@prevcmd\eql@frame@cmd
3465      \tabskip\z@skip
3466      &%
3467      \eql@prevwidth@\wd\eql@cellbox@
3468      \let\eql@box@lastcell\eql@box@lastcell@even
3469      \let\eql@frame@cmd\eql@frame@prevcmd
3470      \global\let\eql@cell@container\@empty
3471      \setbox\eql@cellbox@\hbox{%
3472        \unhbox\eql@cellbox@
3473        \eql@strut@cell
3474        \@lign
3475        $\m@th\displaystyle
3476        \eql@hook@innerright
3477        \eql@class@innerright@sel
3478        ##%
3479        \eql@punct@apply@col
3480        \eql@hook@colout
3481        \eql@tagging@mathsave
3482        $%
3483        \eql@tagging@mathaddlast
3484      }%
3485      \eql@cell@container
3486      \ifdefined\eql@frame@cmd
3487        \eql@frame@measure
3488        \advance\eql@prevwidth@\eql@frame@margin@
3489        \eql@frame@print
3490      \fi
3491      \kern-\eql@prevwidth@
3492      \unhbox\eql@cellbox@
3493      \hfil
3494      \tabskip\eql@colsep@\relax
3495      \crrr
3496      \noalign{%
3497        \eql@hook@blockbefore
3498      }%
3499      \eql@hook@blockin
3500 }

3501 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

11.4 Main

```

3502 \let\eql@box@box\vcenter
3503 \let\eql@box@open\@undefined
3504 \let\eql@box@frame\@firstofone
3505 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

```

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!

```

3506 \def\eql@box@close{%
3507   \ifvmode\else
3508     \global\let\eql@shape@lastrow\eql@true
3509     \eql@punct@apply@block

```



```

3510     \eql@box@cr@
3511     \fi
3512     \noalign{%
3513         \eql@hook@blockafter
3514         \global\let\eql@shape@lastrow\eql@false
3515     }%
3516     \eql@tagging@tablesaveinner
3517 \egroup
3518 }

```

\eql@box@vcenter

```

3519 \def\eql@box@vcenter#1{%
3520     \ifmmode
3521         \vcenter{#1}%
3522     \else
3523         $\m@th\vcenter{#1}$%
3524     \fi
3525 }

```

\eql@box@start

```

3526 \let\eql@box@endmath\eql@false
3527 \def\eql@box@start{%
3528     \relax
3529     \ifmmode
3530         \let\eql@box@endmath\eql@false
3531     \else
3532         \let\eql@box@endmath\eql@true
3533         \expandafter$%$
3534     \fi
3535     \eql@box@processopt
3536     \eql@stack@save@box
3537     \let\eql@frame@cmd\@undefined
3538     \let\eql@layoutleft\eql@false
3539     \eql@row@z@
3540     \eql@totalrows@\@M
3541     \eql@shape@select
3542     \setbox\z@\ifx\eql@box@box\vcenter
3543         \expandafter\vbox
3544     \else
3545         \expandafter\eql@box@box
3546     \fi\bgroup
3547     \eql@display@nest
3548     \let\\eql@box@cr
3549     \eql@spread@set
3550     \eql@strut@make
3551     \eql@box@open
3552 }

```

\eql@box@end

```

3553 \def\eql@box@end{%
3554     \eql@box@close
3555     \egroup
3556     \eql@box@frame{%
3557         \ifdefined\eql@display@marginleft
3558             \hskip\glueexpr\eql@display@marginleft\relax
3559         \fi

```

```

3560 \ifx\eql@box@box\vcenter
3561 \eql@box@vcenter{\unvbox\z@}%
3562 \else
3563 \box\z@
3564 \fi
3565 \eql@tagging@tableaddinner
3566 \ifdefined\eql@display@marginright
3567 \hskip\glueexpr\eql@display@marginright\relax
3568 \fi
3569 }%
3570 \eql@stack@restore
3571 \ifdefined\eql@box@endmath
3572 \expandafter$%$
3573 \fi
3574 }

```

11.5 Environment

`equationsbox` (*env.*)

```

3575 \newenvironment{equationsbox}{%
3576 \eql@verbose@info\eql@verbose@msg@enterenv
3577 \eql@ampprotect\eql@box@testall\eql@box@start
3578 }{%
3579 \eql@box@end
3580 \eql@verbose@info\eql@verbose@msg@leaveenv
3581 }

3582 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3583 \def\eql@box@parseopt{%
3584 \ifx\eql@parseopt@token[%]
3585 \let\eql@parseopt@next\eql@parseopt@opt
3586 \fi
3587 \ifx\eql@parseopt@token=%
3588 \let\eql@parseopt@next\eql@parseopt@lines
3589 \fi
3590 \ifx\eql@parseopt@token|
3591 \let\eql@parseopt@next\eql@parseopt@columns
3592 \fi
3593 \ifx\eql@parseopt@token'
3594 \let\eql@parseopt@next\eql@parseopt@punctall
3595 \fi
3596 }

```

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

```

3597 \def\eql@box@processopt{%
3598 \let\eql@box@frame\@firstofone
3599 \let\eql@display@marginleft\@undefined
3600 \let\eql@display@marginright\@undefined
3601 \eql@nextopt@process{equationsbox}%
3602 \let\eql@punct@block\eql@punct@main
3603 \let\eql@punct@main\relax
3604 \eql@colsep@\glueexpr\eql@box@colsep\relax
3605 \ifdefined\eql@paddingleft@val
3606 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3607 \else
3608 \eql@paddingleft@\z@

```

```

3609 \fi
3610 \ifdefined\eql@paddingright@val
3611 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3612 \else
3613 \eql@paddingright@\z@
3614 \fi
3615 \eql@indent@\glueexpr\eql@indent@val\relax
3616 }

```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

3617 \def\eql@single@start@native{%
3618 \eql@display@init
3619 \eql@display@print
3620 \let\raisetag\eql@raisetag@default
3621 \eql@shape@align@disable
3622 \eql@hook@eqin
3623 % \mathopen{}%
3624 }%

```

TODO: describe

```

3625 \def\eql@single@end@native{%
3626 % \mathclose{}%
3627 \eql@tags@container
3628 \eql@numbering@single@eval
3629 \if@eqnsw
3630 \ifdefined\eql@tagsleft
3631 \leqno
3632 \else
3633 \eqno
3634 \fi
3635 \eql@composetag@print
3636 \fi
3637 \eql@interline@container
3638 \advance\eql@belowspace@\eql@vspaceskip@
3639 \eql@display@container
3640 \eql@display@penalty
3641 \eql@display@vspace@native
3642 }%

```

12.2 Print

```

3643 \def\eql@single@start@print{%
3644 \eql@display@init
3645 \eql@display@print
3646 \eql@shape@align@enable
3647 \eql@totalrows@\@ne
3648 \eql@row@\@ne
3649 \eql@arrange@init
3650 \global\let\eql@cell@container\@empty
3651 \prevgraf\numexpr\prevgraf+\@ne\relax
3652 \setbox\eql@cellbox@\hbox\bgroup

```

```

3653 \eq restore@hfuzz
3654 \eq strut@cell
3655 $\m@th\displaystyle%$
3656 \eq hook@eqin
3657 }

3658 \def\eq@single@end@print{%
3659 \eq@tagging@mathsave
3660 $%$
3661 \hfil
3662 \kern\z@
3663 \egroup
3664 \prevgraf\numexpr\prevgraf-\@ne\relax

3665 \eq@shape@eval
3666 \eq@cell@container

3667 \ifdefined\eq@frame@cmd
3668 \eq@frame@adjust
3669 \fi

3670 \eq@cellwidth@\wd\eq@cellbox@
3671 \eq@line@height@\ht\eq@cellbox@
3672 \eq@line@depth@\dp\eq@cellbox@
3673 \eq@totalwidth@\eq@cellwidth@
3674 \eq@totalheight@\dimexpr\eq@line@height@+\eq@line@depth@\relax
3675 \eq@topheight@\eq@line@height@
3676 \eq@bottomdepth@\eq@line@depth@

3677 \eq@tags@container
3678 \eq@numbering@single@eval
3679 \if@eqnsw
3680 \eq@tagbox@make\eq@composetag@print
3681 \eq@tagrows@\@ne
3682 \ifdefined\eq@tagpos@reserve\else
3683 \eq@tagwidth@\z@
3684 \fi
3685 \eq@tagheight@block@\ht\eq@tagbox@
3686 \eq@tagdepth@block@\dp\eq@tagbox@
3687 \else
3688 \eq@numbering@warnunused
3689 \eq@tagwidth@\z@
3690 \eq@tagrows@\z@
3691 \fi
3692 \eq@tagwidth@max@\eq@tagwidth@
3693 \eq@tagpos@single@eval
3694 \eq@tagpos@print@line@eval

3695 \eq@intercolumns@\z@
3696 \eq@adjust@calc@lines

3697 \eq@display@halign@init{}%
3698 \halign{##\crr
3699 \noalign{\eq@display@halign@start}%
3700 \eq@arrange@print@line
3701 \cr
3702 \noalign{\eq@display@halign@end}%
3703 \eq@tagging@tablesavelines
3704 }%
3705 \eq@tagpos@print@line@end
3706 \eq@display@close
3707 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```
3708 \def\eql@lines@measure@line@begin{%
3709   \eql@verbose@info\eql@verbose@msg@startline
3710   \eql@numbering@measure@line@begin
3711   \eql@hook@linein
3712 }
```

TODO: describe

```
3713 \def\eql@lines@measure@line@end{%
3714   \eql@punct@apply@line
3715   \eql@hook@lineout
3716 }
```

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

```
3717 \def\eql@lines@measure@cell{%
3718   \ifdefined\eql@frame@cmd
3719     \ifcase\eql@shape@pos@
3720       \eql@frame@measure
3721       \advance\eql@shape@amount@-\eql@frame@margin@
3722     \or\or
3723       \eql@frame@measure
3724       \advance\eql@shape@amount@+\eql@frame@margin@
3725     \fi
3726     \eql@frame@print
3727   \fi
3728   \eql@cellwidth@\wd\eql@cellbox@
3729   \eql@line@height@\ht\eql@cellbox@
3730   \eql@line@depth@\dp\eql@cellbox@
3731   \eql@dimensions@startrow
3732   \eql@dimensions@savecell
3733   \kern\eql@cellwidth@
3734 }
```

\eql@lines@measure

```
3735 \def\eql@lines@measure{%
3736   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
3737   \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3738   \eql@totalrows@\@M
3739   \eql@shape@select

3740   \setbox\z@\vbox{\measuring@true\halign{%
3741     \global\let\eql@cell@container\@empty
3742     \setbox\eql@cellbox@\hbox{%
3743       \eql@strut@cell
3744       \@lign
3745       $\m@th\displaystyle
3746       \eql@hook@colin
3747       ##%
```

```

3748         \eql@punct@apply@col
3749         \eql@hook@colout
3750     $%
3751 }%
3752 \ifdefined\eql@shape@lastrow
3753     \eql@totalrows@\eql@row@
3754 \fi
3755 \eql@shape@eval
3756 \eql@cell@container
3757 \eql@lines@measure@cell
3758 \eql@measure@tag
3759 \eql@measure@endrow
3760 \crrr

3761 \noalign{%
3762     \global\let\eql@shape@lastrow\eql@false
3763     \eql@hook@blockbefore
3764 }%
3765 \eql@hook@blockin
3766 \eql@scan@body
3767 \ifvmode\else
3768     \global\let\eql@shape@lastrow\eql@true
3769     \eql@punct@apply@block
3770     \eql@hook@blockout
3771     \eql@display@endline
3772 \cr
3773 \fi
3774 \omit
3775 \cr
3776 \noalign{%
3777     \eql@hook@blockafter
3778     \global\let\eql@shape@lastrow\eql@false
3779 }%
3780 }}%

3781 \eql@measure@close

3782 \setbox\z@\vbox{%
3783     \unvbox\z@
3784     \unpenalty
3785     \global\setbox\@ne\lastbox
3786 }%
3787 \eql@totalwidth@\wd\@ne

3788 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@measure
3789 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

3790 \def\eql@lines@adjust{%
3791     \eql@tagpos@adjust@eval
3792     \eql@adjust@calc@lines
3793     \eql@numbering@best@eval
3794 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```
3795 \def\eq@lines@print@line@begin{%
3796   \eq@verbose@info\eq@verbose@msg@startline
3797   \eq@numbering@print@line@begin
3798   \eq@hook@linein
3799 }
```

TODO: describe

```
3800 \def\eq@lines@print@line@end{%
3801   \eq@punct@apply@line
3802   \eq@hook@lineout
3803 }
```

TODO: describe

```
3804 \def\eq@lines@print@line@adjust{%
3805   \ifdefined\eq@frame@cmd
3806     \ifcase\eq@shape@pos@
3807       \eq@frame@measure
3808       \advance\eq@shape@amount@-\eq@frame@margin@
3809     \or\or
3810       \eq@frame@measure
3811       \advance\eq@shape@amount@+\eq@frame@margin@
3812     \fi
3813     \eq@frame@adjust
3814   \fi
3815   \eq@cellwidth@\wd\eq@cellbox@
3816   \eq@line@height@\ht\eq@cellbox@
3817   \eq@line@depth@\dp\eq@cellbox@
3818   \eq@numbering@print@line@eval
3819   \if@eqnsw
3820     \eq@tagbox@make\eq@composetag@print
3821   \fi
3822   \eq@tagpos@print@line@eval
3823   \eq@arrange@print@line
3824   \eq@tagpos@print@line@end
3825 }
```

TODO: describe

```
3826 \def\eq@lines@print{%
3827   \eq@verbose@infoarg\eq@verbose@msg@center\eq@lines@print
3828   \eq@arrange@init
3829   \eq@display@halign@init\eq@lines@print@line@begin
3830   \eq@display@halign@letcr\eq@lines@print@line@end
3831   \tabskip\z@skip

3832   \halign{%
3833     \global\let\eq@cell@container\@empty
3834     \setbox\eq@cellbox@\hbox{%
3835       \eq@restore@hfuzz
3836       \eq@strut@cell
3837       \@lign
3838       $\m@th\displaystyle
3839       \eq@hook@colin
```

```

3840      ##%
3841      \eql@punct@apply@col
3842      \eql@hook@colout
3843      \eql@tagging@mathsave
3844      $%
3845      \hfil
3846      \kern\z@
3847      }%
3848      \eql@shape@eval
3849      \eql@cell@container
3850      \eql@lines@print@line@adjust
3851      \crrr

3852      \noalign{%
3853      \eql@display@halign@start
3854      \eql@numbering@print@block@begin
3855      \eql@hook@blockbefore
3856      }%
3857      \eql@hook@blockin
3858      \eql@scan@body
3859      \ifvmode\else
3860      \relax
3861      \eql@punct@apply@block
3862      \eql@hook@blockout
3863      \eql@display@endline
3864      \cr
3865      \fi
3866      \noalign{%
3867      \eql@hook@blockafter
3868      \eql@display@halign@end
3869      \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@print
3870      }%
3871      \eql@tagging@tablesavelines
3872      }%
3873 }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```

\begin{columns}
\columns@add@amp
\columns@completerow
3874 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3875 \def\eql@columns@completerow{%
3876   \count@=\numexpr\eql@totalcolumns@+\@ne-\eql@column@ \relax
3877   \edef\eql@tmp{%
3878     \expandafter\eql@columns@add@amp\romannumeral\count@ 000q}%
3879   \eql@tmp
3880 }

3881 \def\eql@columns@overfull{%
3882   \dimen@=\eql@line@width@

```



```

3883 \advance\dimen@-\hfuzz
3884 \ifdim\dimen@>\displaywidth
3885   \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3886   \wd\z@\z@
3887   \ht\z@\eql@line@height@
3888   \dp\z@\eql@line@depth@
3889   \box\z@
3890 \fi
3891 }

```

14.2 Transpose

TODO: describe

TODO: describe

```

3892 \let\eql@transpose@active\eql@false
3893 \def\eql@transpose@end{\eql@transpose@end}
3894 \def\eql@transpose@skip{&\eqnpunct{}}
3895 \def\eql@transpose@complete{%
3896   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

TODO: describe

```

3897 \def\eql@transpose{%
3898   \eql@totalcolumns@\z@
3899   \eql@totalrows@\z@
3900   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
3901   \eql@scan@reg@{}%
3902   \eql@row@\z@
3903   \eql@transpose@output@row
3904 }

```

TODO: describe

```

3905 \def\eql@transpose@save@col#1{%
3906   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
3907     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

TODO: describe

```

3908 \def\eql@transpose@scan@col#1&{%
3909   \def\eql@tmpa{#1}%
3910   \ifx\eql@tmpa\eql@transpose@end\else
3911     \advance\eql@totalcolumns@\@ne
3912     \eql@row@\z@
3913     \let\eql@transpose@data@col\@empty
3914     \eql@transpose@scan@row#1\\eql@transpose@end\\%
3915     \ifnum\eql@row@>\eql@totalrows@
3916       \eql@totalrows@\eql@row@
3917     \fi
3918     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
3919     \expandafter\eql@transpose@scan@col
3920 \fi
3921 }

```

TODO: describe

```

3922 \def\eql@transpose@append@row#1{%
3923   \advance\eql@row@\@ne
3924   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}

```

TODO: describe

```
3925 \def\eql@transpose@scan@row#1\\{%
3926   \def\eql@tmpa{#1}%
3927   \ifx\eql@tmpa\eql@transpose@end\else
3928     \ifx\eql@transpose@active+
3929       \eql@transpose@scan@cell#1&\eql@transpose@end&%
3930     \else
3931       \eql@transpose@append@row{#1}%
3932     \fi
3933     \expandafter\eql@transpose@scan@row
3934   \fi
3935 }
```

TODO: describe

```
3936 \def\eql@transpose@scan@cell#1&#2&{%
3937   \def\eql@tmpa{#2}%
3938   \ifx\eql@tmpa\eql@transpose@end
3939     \eql@transpose@append@row{#1}%
3940   \else
3941     \eql@transpose@append@row{#1&#2}%
3942     \expandafter\eql@transpose@scan@cell@next
3943   \fi
3944 }
```

TODO: describe

```
3945 \def\eql@transpose@scan@cell@next#1&{%
3946   \def\eql@tmpa{#1}%
3947   \ifx\eql@tmpa\eql@transpose@end\else
3948     \eql@transpose@append@row{&#1}%
3949     \expandafter\eql@transpose@scan@cell@next
3950   \fi
3951 }
```

TODO: describe

```
3952 \def\eql@transpose@output@row{%
3953   \ifnum\eql@row@<\eql@totalrows@
3954     \advance\eql@row@\@ne
3955     \eql@column@\z@
3956     \eql@transpose@output@col
3957     \ifnum\eql@row@<\eql@totalrows@
3958       \eql@scan@addto\\%
3959     \fi
3960     \expandafter\eql@transpose@output@row
3961   \fi
3962 }
```

TODO: describe

```
3963 \def\eql@transpose@output@col{%
3964   \ifnum\eql@column@<\eql@totalcolumns@
3965     \advance\eql@column@\@ne
3966     \csname eql@transpose@data@col@\the\eql@column@\endcsname
3967     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
3968     \ifnum\eql@column@<\eql@totalcolumns@
3969       \eql@scan@addto{\eql@transpose@complete}%
3970     \fi
3971     \expandafter\eql@transpose@output@col
```

```

3972 \fi
3973 }

```

14.3 Measure

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

```

s@measure@line@begin

```

```

3974 \def\eql@columns@measure@line@begin{%
3975   \eql@verbose@info\eql@verbose@msg@startline
3976   \global\eql@column@\z@
3977   \global\eql@line@height@\z@
3978   \global\eql@line@depth@\z@
3979   \eql@numbering@measure@line@begin
3980   \eql@hook@linein
3981 }

```

```

3982 \def\eql@columns@measure@cell{%
3983   \eql@cellwidth@\wd\eql@cellbox@
3984   \ifdefined\eql@frame@cmd
3985     \eql@frame@measure
3986     \advance\eql@cellwidth@\eql@frame@margin@
3987   \fi
3988   \ifdim\ht\eql@cellbox@>\eql@line@height@
3989     \global\eql@line@height@\ht\eql@cellbox@
3990   \fi
3991   \ifdim\dp\eql@cellbox@>\eql@line@depth@
3992     \global\eql@line@depth@\dp\eql@cellbox@
3993   \fi
3994   \ifnum\eql@column@=\@ne
3995     \eql@dimensions@startrow
3996   \fi
3997   \ifodd\eql@column@
3998     \eql@shape@pos@\tw@
3999   \else
4000     \eql@shape@pos@\z@
4001   \fi
4002   \eql@shape@amount@\z@
4003   \eql@dimensions@savecell
4004   \ifodd\eql@column@\else
4005     \eql@dimensions@savesep
4006   \fi
4007   \kern\eql@cellwidth@
4008 }

```

```

mns@measure@line@end

```

```

4009 \def\eql@columns@measure@line@end{%
4010   \eql@punct@apply@line
4011   \eql@hook@lineout
4012   &\omit
4013   \ifnum\eql@column@>\eql@totalcolumns@
4014     \global\eql@totalcolumns@\eql@column@
4015   \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

```

4016 \ifdefined\eqL@frame@cmd
4017   \advance\eqL@column@\@ne
4018   \wd\eqL@cellbox@\z@
4019   \eqL@columns@measure@cell
4020 \fi
4021 \eqL@measure@tag
4022 \eqL@measure@endrow
4023 }

```

\eqL@columns@measure

```

4024 \def\eqL@columns@measure{%
4025   \eqL@verbose@infoarg\eqL@verbose@msg@enter\eqL@columns@measure
4026   \eqL@totalcolumns@\z@
4027   \eqL@measure@init\eqL@columns@measure@line@begin\eqL@columns@measure@line@end

4028   \setbox\z@\vbox{\measuring@true\halign{%
4029     &%
4030     \global\advance\eqL@column@\@ne
4031     \global\let\eqL@cell@container\@empty
4032     \global\setbox\eqL@cellbox@\hbox{%
4033       \eqL@strut@cell
4034       \@lign
4035       $\m@th\displaystyle
4036       \eqL@hook@colin
4037       ##%
4038       \eqL@class@innerleft
4039       \eqL@hook@innerleft
4040       $%
4041     }%
4042     \eqL@cell@container
4043     \hfil
4044     \eqL@columns@measure@cell
4045     \global\let\eqL@frame@prevcmd\eqL@frame@cmd
4046     &%
4047     \eqL@prevwidth@\wd\eqL@cellbox@
4048     \let\eqL@frame@cmd\eqL@frame@prevcmd
4049     \global\advance\eqL@column@\@ne
4050     \global\let\eqL@cell@container\@empty
4051     \setbox\eqL@cellbox@\hbox{%
4052       \eqL@strut@cell
4053       \@lign
4054       $\m@th\displaystyle
4055       \eqL@hook@innerright
4056       \eqL@class@innerright@sel
4057       ##%
4058       \eqL@punct@apply@col
4059       \eqL@hook@colout
4060       $%
4061     }%
4062     \eqL@cell@container
4063     \eqL@columns@measure@cell
4064     \hfil
4065   \crcr

4066   \noalign{%
4067     \eqL@hook@blockbefore

```

```

4068 }%
4069 \eq@hook@blockin
4070 \eq@scan@body

4071 \ifvmode\else
4072 \eq@punct@apply@block
4073 \eq@hook@blockout
4074 \eq@display@endline
4075 \cr
4076 \fi
4077 \noalign{%
4078 \eq@hook@blockafter
4079 }%

```

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

```

4080 \omit
4081 \eq@column@\@ne
4082 \eq@columns@completerow
4083 \cr
4084 }}%

4085 \eq@measure@close

4086 \setbox\z@\vbox{%
4087 \unvbox\z@
4088 \unpenalty
4089 \global\setbox\@ne\lastbox
4090 }%
4091 \eq@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4092 \let\eq@colwidth@tab\@empty
4093 \loop
4094 \setbox\@ne\hbox{%
4095 \unhbox\@ne
4096 \unskip
4097 \global\setbox\thr@@\lastbox
4098 }%
4099 \ifhbox\thr@@
4100 \eq@colwidth@save{\wd\thr@@}%
4101 \repeat

4102 \eq@verbose@infoarg\eq@verbose@msg@leave\eq@columns@measure
4103 }

```

14.4 Columns Placement

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

```

4104 \def\eq@columns@adjust{%
4105 \ifodd\eq@totalcolumns@
4106 \advance\eq@totalcolumns@\@ne
4107 \fi
4108 \eq@tagpos@adjust@eval
4109 \eq@adjust@calc@columns
4110 }

```

14.5 Print

TODO: describe

mns@print@line@begin

```
4111 \def\eq@columns@print@line@begin{%
4112   \eq@verbose@info\eq@verbose@msg@startline
4113   \global\eq@column@\z@
4114   \global\eq@line@pos@\eq@marginleft@
4115   \global\eq@line@width@\z@
4116   \global\eq@line@avail@\eq@totalwidth@
4117   \global\eq@line@height@\z@
4118   \global\eq@line@depth@\z@
4119   \eq@numbering@print@line@begin
4120   \eq@hook@linein
4121 }
```

l@columns@print@cell

```
4122 \def\eq@columns@print@cell{%
4123   \eq@cellwidth@\wd\eq@cellbox@
4124   \ifodd\eq@column@
4125     \ifdefined\eq@frame@cmd
4126       \eq@frame@measure
4127       \advance\eq@cellwidth@\eq@frame@margin@
4128     \fi
4129     \dimen@\z@
4130   \else
4131     \advance\eq@cellwidth@-\eq@prevwidth@
```

draw a frame

```
4132     \ifdefined\eq@frame@cmd
4133       \eq@frame@measure
4134       \advance\eq@cellwidth@\eq@frame@margin@
4135       \advance\eq@prevwidth@\eq@frame@margin@
4136       \eq@frame@print
4137     \fi
```

update height and depth

```
4138     \ifdim\ht\eq@cellbox@>\eq@line@height@
4139       \global\eq@line@height@\ht\eq@cellbox@
4140     \fi
4141     \ifdim\dp\eq@cellbox@>\eq@line@depth@
4142       \global\eq@line@depth@\dp\eq@cellbox@
4143     \fi
```

print box

```
4144     \kern-\eq@prevwidth@
4145     \unhbox\eq@cellbox@
4146     \dimen@-\eq@cellwidth@
4147   \fi
```

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

```
4148   \advance\dimen@\eq@colwidth@get\eq@column@\relax
4149   \kern\dimen@
```

update available and used space

```

4150 \dimen@eql@colwidth@get\eql@column@\relax
4151 \ifdim\eql@cellwidth@>\z@
4152   \ifdim\eql@line@width@=\z@
4153     \eql@line@avail@\eql@line@pos@
4154     \ifodd\eql@column@
4155       \advance\eql@line@avail@\dimen@
4156       \advance\eql@line@avail@-\eql@cellwidth@
4157     \fi
4158     \global\eql@line@avail@\eql@line@avail@
4159   \fi
4160   \eql@line@width@\eql@line@pos@
4161   \ifodd\eql@column@
4162     \advance\eql@line@width@\dimen@
4163   \else
4164     \advance\eql@line@width@\eql@cellwidth@
4165   \fi
4166   \global\eql@line@width@\eql@line@width@
4167 \fi
4168 \advance\eql@line@pos@\dimen@
4169 \ifodd\eql@column@\else
4170   \advance\eql@line@pos@\eql@colsep@
4171 \fi
4172 \global\eql@line@pos@\eql@line@pos@
4173 }

4174 \def\eql@columns@print@trailright{%
4175   &\omit
4176   \eql@prevwidth@\wd\eql@cellbox@
4177   \let\eql@frame@cmd\eql@frame@prevcmd
4178   \global\advance\eql@column@\@ne
4179   \eql@columns@print@cell
4180 }

```

lums@print@line@end

```

4181 \def\eql@columns@print@line@end{%
4182   \eql@punct@apply@line
4183   \eql@hook@lineout
4184 % \TODO add an even column with empty stuff if box processing deferred
4185   \ifodd\eql@column@
4186     \expandafter\eql@columns@print@trailright
4187   \fi
4188   \eql@columns@completerow
4189   \eql@columns@print@tag
4190 }

```

ql@columns@print@tag

```

4191 \def\eql@columns@print@tag{%
4192   \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4193 \eql@display@firstavail@set\eql@line@avail@
4194 \eql@columns@overfull
4195 \eql@numbering@print@line@eval
4196 \if@eqnsw
4197   \eql@tagbox@make\eql@composetag@print

```

```

4198 \fi
4199 \eql@tagpos@print@line@eval
4200 \eql@tagbox@print@cell
4201 \eql@tagpos@print@line@end
4202 }

```

\eql@columns@print

```

4203 \def\eql@columns@print{%
4204 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
4205 \eql@shape@align@disable
4206 \eql@display@halign@init\eql@columns@print@line@begin
4207 \eql@display@halign@letcr\eql@columns@print@line@end
4208 \tabskip\eql@marginleft@

4209 \halign{%
4210 &%
4211 \global\advance\eql@column@ \@ne
4212 \global\let\eql@cell@container \@empty
4213 \global\setbox\eql@cellbox@ \hbox{%
4214 \eql@strut@cell
4215 \@lign
4216 $\m@th\displaystyle
4217 \eql@hook@colin
4218 ##%
4219 \eql@class@innerleft
4220 \eql@hook@innerleft
4221 \eql@tagging@mathsave
4222 $%
4223 \eql@tagging@mathaddlast
4224 }%
4225 \eql@cell@container
4226 \hfil
4227 \eql@columns@print@cell
4228 \global\let\eql@frame@prevcmd\eql@frame@cmd
4229 \tabskip\z@skip
4230 &%
4231 \eql@prevwidth@\wd\eql@cellbox@
4232 \let\eql@frame@cmd\eql@frame@prevcmd
4233 \global\advance\eql@column@ \@ne
4234 \global\let\eql@cell@container \@empty
4235 \setbox\eql@cellbox@ \hbox{%
4236 \unhbox\eql@cellbox@
4237 \eql@strut@cell
4238 \@lign
4239 $\m@th\displaystyle
4240 \eql@hook@innerright
4241 \eql@class@innerright@sel
4242 ##%
4243 \eql@punct@apply@col
4244 \eql@hook@colout
4245 \eql@tagging@mathsave
4246 $%
4247 \eql@tagging@mathaddlast
4248 }%
4249 \eql@cell@container
4250 \eql@columns@print@cell
4251 \hfil
4252 \tabskip\eql@colsep@\relax

```



```

4253 \crrr

4254 \noalign{%
4255 \eqldisplay@halign@start
4256 \eqlnumbering@print@block@begin
4257 \eqlhook@blockbefore
4258 }%
4259 \eqlhook@blockin
4260 \eqlscan@body
4261 \ifvmode\else
4262 \relax
4263 \eqlpunct@apply@block
4264 \eqlhook@blockout
4265 \eqldisplay@endline
4266 \cr
4267 \fi
4268 \noalign{%
4269 \eqlhook@blockafter
4270 \eqldisplay@halign@end
4271 \eqlverbose@infoarg\eqlverbose@msg@leave\eql@columns@print
4272 }%
4273 \eql@tagging@tables@savealign
4274 }%
4275 }

```

15 Interface

15.1 Scanning the Equation Body

The multi-line equatiuon 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.

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

```
4276 \newtoks\eqlscan@reg@
```

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

```

4277 \def\eqlscan@body@dump{\the\eqlscan@reg@}
4278 \def\eqlscan@body@rescan{%
4279 \expandafter\scantokens\expandafter{\the\eqlscan@reg@}}
4280 \let\eqlscan@body\eqlscan@body@dump

```

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

```
4281 \long\def\eqlscan@addto#1{\eqlscan@reg@\expandafter{\the\eqlscan@reg@#1}}
```

Scan Modifiers at End.

`\eql@scan@testend` Scan for modifiers following the end of the scanned block:

```
4282 \def\eql@scan@testend{%
4283   \eql@ampprotect\eql@equations@testend\eql@scan@end}
```

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@`:

```
4284 \def\eql@scan@env#1{%
4285   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@env
4286   \def\eql@scan@end{#1\expandafter\end\expandafter{\@currentvir}}%
4287   \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`:

```
4288   \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currentvir\endcsname}%
4289   \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
4290   \ifdefined\eql@scan@par
4291     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
4292   \else
4293     \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
4294   \fi
4295   \eql@scan@iterate
4296 }
```

`\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:

```
4297 \long\def\eql@scan@env@iterate#1\end#2{%
4298   \edef\eql@scan@stack{%
4299     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4300   \ifx\@empty\eql@scan@stack
4301     \@checkend{#2}%
4302     \eql@scan@addto{#1}%
4303     \expandafter\let\eql@scan@iterate\eql@scan@env@org
4304     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@env
4305     \expandafter\eql@scan@testend
4306   \else
4307     \eql@scan@addto{#1\end{#2}}%
4308     \expandafter\eql@scan@iterate
4309   \fi
4310 }
```

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

```
4311 \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.

```
4312 \long\def\eql@scan@env@count#1\begin#2{%
4313   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4314 }
```

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

```
4315 \def\eql@scan@env@cancel{%
4316   \@namedef{end\@currentvir}{\ignorespacesafterend}%
4317 }
```

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 `\[`:

```
4318 \def\eql@scan@sqr#1{%
4319   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@sqr
4320   \def\eql@scan@end{#1}}%
4321   \eql@scan@reg@{\def\eql@scan@stack{b}}%
4322   \let\eql@scan@sqr@org\[%\]
4323   \ifdefined\eql@scan@par
4324     \let\[\eql@scan@sqr@iterate%\]
4325   \else
4326     \let\[\eql@scan@sqr@iterate@nopar%\]
4327   \fi
4328   \[%\]
4329 }
```

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

```
4330 \long\def\eql@scan@sqr@iterate#1\[%\]
4331   \edef\eql@scan@stack{%
4332     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4333   \ifx\@empty\eql@scan@stack
4334     \let\[\eql@scan@sqr@org%\]
4335     \eql@scan@addto{#1}%
4336     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@sqr
4337     \expandafter\eql@scan@testend
4338   \else
4339     \eql@scan@addto{#1}}%
4340   \expandafter\[%\]
4341   \fi
4342 }
```

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

```
4343 \def\eql@scan@sqr@iterate@nopar#1\]{\eql@scan@sqr@iterate#1\}}
```

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

```

4344 \long\def\eql@scan@sqr@count#1\[#2{\%\\
4345   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4346 }

```

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

```

4347 \def\eql@scan@sqrang@cancel{\expandafter\ignorespaces\@gobble}

```

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

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

```

4348 \def\eql@scan@ang#1{%
4349   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@ang
4350   \def\eql@scan@end{#1\>}%
4351   \eql@scan@reg@{\}\def\eql@scan@stack{b}%
4352   \let\eql@scan@ang@org\<%\>
4353   \ifdefined\eql@scan@par
4354     \let\<\eql@scan@ang@iterate%\>
4355   \else
4356     \let\<\eql@scan@ang@iterate@nopar%\>
4357   \fi
4358   \<%\>
4359 }

```

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

```

4360 \long\def\eql@scan@ang@iterate#1\>{%
4361   \edef\eql@scan@stack{%
4362     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4363   \ifx\@empty\eql@scan@stack
4364     \let\<\eql@scan@ang@org%\>
4365     \eql@scan@addto{#1}%
4366     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@ang
4367     \expandafter\eql@scan@testend
4368   \else
4369     \eql@scan@addto{#1\>}%
4370     \expandafter\<%\>
4371   \fi
4372 }

```

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

```

4373 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}

```

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

```

4374 \long\def\eql@scan@ang@count#1\<#2{\%\\
4375   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4376 }

```

15.2 Options Processing

`\eql@equations@testall` The macro sequence started by `\eql@equations@testall` scans for optional arguments to the equation environments and appends them to the argument list using `\eqnaddopt`. All

arguments are scanned such that any spaces stop the scanning and such that any alignment markers ‘&’ cannot interfere: **TODO:** update

```

4377 \def\eq@equations@testall{\eq@parseopt\eq@equations@parseopt}
4378 \def\eq@equations@parseopt{%
4379   \ifx\eq@parseopt@token*%
4380     \let\eq@parseopt@next\eq@parseopt@nonumber
4381   \fi
4382   \ifx\eq@parseopt@token!%
4383     \let\eq@parseopt@next\eq@parseopt@donumber
4384   \fi
4385   \ifx\eq@parseopt@token/%
4386     \let\eq@parseopt@next\eq@parseopt@transpose
4387   \fi
4388   \ifx\eq@parseopt@token[%]
4389     \let\eq@parseopt@next\eq@parseopt@opt
4390   \fi
4391   \ifx\eq@parseopt@token\eq@atxi
4392     \let\eq@parseopt@next\eq@parseopt@label
4393   \fi
4394   \ifx\eq@parseopt@token\eq@atxii
4395     \let\eq@parseopt@next\eq@parseopt@label
4396   \fi
4397   \ifx\eq@parseopt@token.%
4398     \let\eq@parseopt@next\eq@parseopt@punctdot
4399   \fi
4400   \ifx\eq@parseopt@token,%
4401     \let\eq@parseopt@next\eq@parseopt@punctcomma
4402   \fi
4403   \ifx\eq@parseopt@token~%
4404     \let\eq@parseopt@next\eq@parseopt@punctoff
4405   \fi
4406   \ifx\eq@parseopt@token'%
4407     \let\eq@parseopt@next\eq@parseopt@punctall
4408   \fi
4409   \ifx\eq@parseopt@token-%
4410     \let\eq@parseopt@next\eq@parseopt@single
4411   \fi
4412   \ifx\eq@parseopt@token=%
4413     \let\eq@parseopt@next\eq@parseopt@lines
4414   \fi
4415   \ifx\eq@parseopt@token|%
4416     \let\eq@parseopt@next\eq@parseopt@columns
4417   \fi
4418   \ifx\eq@parseopt@token\label
4419     \let\eq@parseopt@next\eq@parseopt@end
4420   \fi
4421   \ifx\eq@parseopt@token\begin
4422     \let\eq@parseopt@next\eq@parseopt@end
4423   \fi
4424 }
```

q1@equations@testend **TODO:** describe

```

4425 \def\eq@equations@testend{\eq@parseopt\eq@equations@parseend}
4426 \def\eq@equations@parseend{%
4427   \ifx\eq@parseopt@token.%
4428     \let\eq@parseopt@next\eq@parseopt@enddot
4429   \fi
```

```

4430 \ifx\eql@parseopt@token,%
4431   \let\eql@parseopt@next\eql@parseopt@endcomma
4432 \fi
4433 \ifx\eql@parseopt@token~%
4434   \let\eql@parseopt@next\eql@parseopt@endoff
4435 \fi
4436 \ifx\eql@parseopt@token'%
4437   \let\eql@parseopt@next\eql@parseopt@endpar
4438 \fi
4439 }

```

`\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:

```

4440 \def\eql@equations@processopt{%
4441   \let\eql@tags@container@block\eql@tags@container@clear
4442   \let\eql@tags@frame@cmd\@firstofone
4443   \let\eql@skip@force@above\@undefined
4444   \let\eql@skip@force@below\@undefined
4445   \let\eql@skip@force@leave\@undefined
4446   \let\eql@display@linewidth\@undefined
4447   \let\eql@display@marginleft\@undefined
4448   \let\eql@display@marginright\@undefined
4449   \eql@abovespace@z@skip
4450   \eql@belowspace@z@skip
4451   \eql@displaybreak@prepen@\@MM
4452   \eql@displaybreak@postpen@\@MM
4453   \eql@nextopt@process{equations}%
4454   \let\eql@punct@block\eql@punct@main
4455   \let\eql@punct@main\relax
4456   \eql@indent@\glueexpr\eql@indent@val\relax
4457   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4458 }

```

15.3 Single-Line Main

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

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

```

4459 \def\eql@single@cr@error{%
4460   \eql@error{Cannot use '\string\\' within display equation.
4461     Please switch to equations environment}%
4462 }

```

`\eql@single@start` Opening code for single-line equation. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, set native vs. manual equation tag mode, install error message for using `\`. Hand over to mode-specific opening:

```

4463 \def\eql@single@start{%
4464   \eql@display@enter
4465   \eql@tagging@start
4466   \eql@dollar@dollar@begin
4467   \eql@display@adjust

```

```

4468 \eql@numbering@init
4469 \eql@stack@save@equations
4470 \eql@numbering@single@init
4471 \ifdefined\eql@single@cr@mode
4472   \let\\\eql@single@cr@mode
4473 \fi
4474 \ifdefined\eql@single@native
4475   \let\eql@single@start@sel\eql@single@start@native
4476   \let\eql@single@end@sel\eql@single@end@native
4477 \else
4478   \let\eql@single@start@sel\eql@single@start@print
4479   \let\eql@single@end@sel\eql@single@end@print
4480 \fi
4481 \eql@single@start@sel
4482 }

```

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

```

4483 \def\eql@single@end{%
4484   \eql@punct@apply@block
4485   \eql@hook@eqout
4486   \eql@single@end@sel
4487   \eql@stack@restore
4488   \eql@dollar@dollar@end
4489   \eql@tagging@end
4490   \eql@display@leave
4491 }

```

`\eql@single@main` Combined opening, body and closing for pre-scanned body: **TODO:** is `\expandafter` needed? relic?

```

4492 \def\eql@single@main{%
4493   \expandafter\eql@single@start
4494   \eql@scan@body
4495   \eql@single@end
4496 }

```

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

```

4497 \def\eql@mode@single{%
4498   \ifdefined\eql@single@doscan
4499     \let\eql@equations@main\eql@single@main
4500     \let\eql@equations@end@empty
4501   \else
4502     \let\eql@equations@main@undefined
4503     \let\eql@equations@end\eql@single@end
4504   \fi
4505 }

```

15.4 Multi-Line Main

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

```

4506 \let\eql@multi@mode@lines\eql@false

```

`\eql@multi@main` Main routine for multi-line modes. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global

registers, initialise macros for use within equations: **TODO:** shove depends on lines vs columns

```

4507 \def\eq\@multi\@main{%
4508   \eq\@display\@enter
4509   \eq\@tagging\@start
4510   \eq\@dollar\@begin
4511   \eq\@display\@adjust
4512   \eq\@numbering\@init
4513   \eq\@stack\@save\@equations
4514   \ifdefined\eq\@transpose\@active
4515     \ifdefined\eq\@multi\@mode\@lines\@else
4516       \eq\@transpose
4517     \fi
4518   \fi
4519   \ifdefined\eq\@numbering\@subeq\@use
4520     \eq\@numbering\@subeq\@init
4521   \fi
4522   \eq\@display\@init
4523   \let\intertext\eq\@intertext
4524   \let\endintertext\endeq\@intertext
4525   \eq\@shape\@align\@enable

```

Now measure the given multi-line equations body:

```

4526   \ifdefined\eq\@multi\@mode\@lines
4527     \eq\@lines\@measure
4528   \else
4529     \ifdefined\eq\@ampproof\@active
4530       \eq\@ampproof
4531     \fi
4532     \eq\@columns\@measure
4533   \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 `\eq\@lines\@measure`:

```

4534   \ifdefined\eq\@numbering\@subeq\@use
4535     \eq\@numbering\@subeq\@test
4536   \fi
4537   \ifdefined\eq\@multi\@mode\@lines\@else
4538     \ifdefined\eq\@multi\@lines\@fallback
4539       \ifnum\eq\@totalcolumns\@=\@one
4540         \let\eq\@multi\@mode\@lines\eq\@true
4541         \ifx\eq\@multi\@lines\@fallback\z@\@else
4542           \eq\@lines\@measure
4543         \fi
4544       \fi
4545     \fi
4546   \fi

```

Adjust the multi-line equations body:

```

4547   \ifdefined\eq\@multi\@mode\@lines
4548     \eq\@lines\@adjust
4549   \else
4550     \eq\@columns\@adjust
4551   \fi

```


Now print the multi-line equations body:

```

4552 \eqldisplay@print
4553 \eql@numbering@print@init
4554 \ifdefined\eql@multi@mode@lines
4555   \eql@lines@print
4556 \else
4557   \eql@columns@print
4558 \fi
4559 \eqldisplay@close

```

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

```

4560 \ifdefined\eql@numbering@subeq@use
4561   \eql@numbering@subeq@close
4562 \fi
4563 \eql@stack@restore
4564 \eqldollardollar@end
4565 \eql@tagging@end
4566 \eqldisplay@leave
4567 }

```

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

```

\eql@mode@lines
4568 \def\eql@mode@columns{%
4569   \let\eqlequations@main\eql@multi@main
4570   \let\eqlequations@end\@empty
4571   \let\eql@multi@mode@lines\eql@false
4572 }
4573 \def\eql@mode@lines{%
4574   \let\eqlequations@main\eql@multi@main
4575   \let\eqlequations@end\@empty
4576   \let\eql@multi@mode@lines\eql@true
4577 }

```

15.5 Equations Environment

We now declare the main environment and its symbolic versions.

Environment.

equations (*env.*) Declare the main equations environment. If already in math mode, fail and cancel the environment body. Otherwise scan for optional arguments and pass on to `\eqlequations@start`:

```

4578 \newenvironment{equations}{%
4579   \eql@verbose@info\eql@verbose@msg@enterenv
4580   \ifmmode
4581     \eql@error@mathmode{\string\begin{\@currenvir}}%
4582     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4583   \else
4584     \expandafter\eql@ampprotect\expandafter\eqlequations@testall
4585     \expandafter\eqlequations@start
4586   \fi
4587 }{%
4588   \eqlequations@end
4589   \ignorespacesafterend

```

```

4590 \eql@verbose@info\eql@verbose@msg@leaveenv
4591 }
4592 \eql@markline@amsthm@register{equations}
4593 \eql@tagging@register@luamml{equations}

```

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

```

4594 \def\eql@equations@start{%
4595   \eql@equations@processopt
4596   \ifdefined\eql@equations@main
4597     \expandafter\eql@scan@env\expandafter\eql@equations@main
4598   \else
4599     \expandafter\eql@single@start
4600   \fi
4601 }

```

Square Brackets.

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

```

4602 \newenvironment{equations@sqr}{}{}
4603 \eql@markline@amsthm@register{equations@sqr}
4604 \eql@tagging@register@luamml{equations@sqr}

```

\eql@equations@sqr@open Definition for ‘`\[`’. If already in math mode, ignore the enclosed contents. Otherwise add the default arguments `\eql@equations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eql@equations@sqr@start`:

```

4605 \protected\def\eql@equations@sqr@open{%
4606   \ifmmode
4607     \eql@error@mathmode{\string\[\dots\string\]}%
4608     \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
4609   \else
4610     \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
4611     \begin{equations@sqr}%
4612     \eql@verbose@info\eql@verbose@msg@enterenv
4613     \let\]\eql@equations@sqr@close
4614     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4615     \expandafter\eql@equations@sqr@start
4616   \fi
4617 }

```

\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`:

```

4618 \def\eql@equations@sqr@start{%
4619   \eql@equations@processopt
4620   \ifdefined\eql@equations@main
4621     \expandafter\eql@scan@sqr\expandafter\eql@equations@main
4622   \else
4623     \expandafter\eql@single@start
4624   \fi
4625 }

```

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

```
4626 \protected\def\eql@equations@sqr@close{%
4627   \eql@ampprotect\eql@equations@testend\eql@equations@sqr@end}
```

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

```
4628 \def\eql@equations@sqr@end{%
4629   \eql@equations@end
4630   \eql@verbose@info\eql@verbose@msg@leaveenv
4631   \end{equations@sqr}%
4632   \ignorespaces
4633 }
```

TODO: describe

```
\eql@sqr@open
\eql@sqr@close
4634 \let\eql@sqr@open\eql@equations@sqr@open
4635 \protected\def\eql@sqr@close{%
4636   \eql@error{'\string\}' may only close '\string\['}%\]
4637 }
```

Angle Brackets.

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

```
4638 \newenvironment{equations@ang}{}{}
4639 \newenvironment{equationsbox@ang}{}{}
4640 \eql@markline@amsthm@register{equations@ang}
4641 \eql@tagging@register@luamml{equations@ang}
```

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

```
4642 \protected\def\eql@ang@open{%
4643   \ifmmode
4644     \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4645     \begin{equationsbox@ang}%
4646     \eql@verbose@info\eql@verbose@msg@enterenv
4647     \let\>\eql@box@ang@close
4648     \expandafter\eql@ampprotect\expandafter\eql@box@testall
4649     \expandafter\eql@box@start
4650   \else
4651     \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
4652     \begin{equations@ang}%
4653     \eql@verbose@info\eql@verbose@msg@enterenv
4654     \let\>\eql@equations@ang@close
4655     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4656     \expandafter\eql@equations@ang@start
4657   \fi
4658 }
```

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

```
4659 \protected\def\eql@ang@close{%
4660   \eql@error{'\string\>'} may only close '\string\<'}%\>
4661 }
```

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

```
4662 \def\eq@equations@ang@start{%
4663   \eq@equations@processopt
4664   \ifdefined\eq@equations@main
4665     \expandafter\eq@scan@ang\expandafter\eq@equations@main
4666   \else
4667     \expandafter\eq@single@start
4668   \fi
4669 }
```

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

```
4670 \def\eq@equations@ang@close{%
4671   \eq@ampprotect\eq@equations@testend\eq@equations@ang@end}
```

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

```
4672 \def\eq@equations@ang@end{%
4673   \eq@equations@end
4674   \eq@verbose@info\eq@verbose@msg@leaveenv
4675   \end{equations@ang}%
4676   \ignorespaces
4677 }
```

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

```
4678 \def\eq@box@ang@close{%
4679   \eq@ampprotect\eq@equations@testend\eq@box@ang@end}
```

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

```
4680 \def\eq@box@ang@end{%
4681   \eq@box@end
4682   \eq@verbose@info\eq@verbose@msg@leaveenv
4683   \end{equationsbox@ang}%
4684   \ignorespaces
4685 }
```

16 Options

16.1 Selection Tools

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

```
4686 \def\eq@decide@abovebelow#1#2#3#4#5{%
4687   \eq@decide@select{#1}{#2}{#3}{%
4688     {,abovebelow,both,tb}{#4#5},%
4689     {above,top,t}{#4},%
4690     {below,bottom,b}{#5}}}
```

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

```
4691 \def\eq@decide@situation#1#2#3#4{%
4692   \eq@decide@select{#1}{#2}{#3}{%
4693     {{long}}{\def#4{0}}},%
4694     {{short}}{\def#4{1}}},%
```

```

4695     {{cont}}{\def#4{2}}},%
4696     {{par}}{\def#4{3}}},%
4697     {{top}}{\def#4{4}}},%
4698     {{noskip}}{\def#4{5}}},%
4699     {{medskip}}{\def#4{6}}}}

```

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:

```

4700 \eqld@define@key{equationsbox}{gathered,gather,ga,lines,ln}[] {%
4701   \eqld@mode@stacked}
4702 \eqld@define@key{equationsbox}{aligned,align,al,columns,col}[] {%
4703   \eqld@mode@aligned}
4704 \eqld@define@key{equationsbox}{top,t}[] {\let\eqld@box@box\vtop}
4705 \eqld@define@key{equationsbox}{center,c}[] {\let\eqld@box@box\vcenter}
4706 \eqld@define@key{equationsbox}{bottom,b}[] {\let\eqld@box@box\vbox}
4707 \eqld@define@key{setup}{boxangopt}[] {%
4708   \def\eqld@box@ang@opt{columns,#1}}

```

Modes for Equations Environment. Declare modes and switches for the equations environment:

```

4709 \eqld@define@key{equations}{equation,eq,single,1}[] {\eqld@mode@single}
4710 \eqld@define@key{equations}{gathered,gather,ga,lines,ln}[] {%
4711   \eqld@mode@lines}
4712 \eqld@define@key{equations}{aligned,align,al,columns,col}[] {%
4713   \eqld@mode@columns}
4714 \eqld@define@key{equations,setup}{transpose}[true] {%
4715   \eqld@decide@select{#3}{#2}{#1} {%
4716     {\eqld@decide@false{\let\eqld@transpose@active\eqld@false}},%
4717     {\noamp,plain,restricted}{\let\eqld@transpose@active\eqld@true}},%
4718     {\eqld@decide@true,amp,cont}{\let\eqld@transpose@active=+}}}%
4719 \eqld@define@key{equations}{native}[true] {%
4720   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native%
4721   \ifdefined\eqld@single@native\let\eqld@layoutleft\eqld@false\fi}
4722 \eqld@define@key{setup}{native}[true] {%
4723   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native}
4724 \eqld@define@key{setup}{scanequation}[true] {%
4725   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@doscan}
4726 \eqld@define@key{setup}{sqropt}[] {%
4727   \def\eqld@equations@sqr@opt{equation,#1}}
4728 \eqld@define@key{setup}{angopt}[] {%
4729   \def\eqld@equations@ang@opt{columns,#1}}

```

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

```

4730 \def\eqld@keycat{equations,equationsbox,setup}
4731 \eqld@define@key\eqld@keycat{spread}{\def\eqld@spread@val{#1}}
4732 \eqld@define@key\eqld@keycat{strut}[true] {\eqld@decide@select{#3}{#2}{#1} {%
4733   {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%
4734   {{cell}}{\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\relax}},%

```

```

4735    {\tag\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
4736    {\eql@decide@true
4737     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
4738 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings concerning page breaks:

```

4739 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4740   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
4741   {{high,3}{\eql@displaybreak@pre3}},%
4742   {{med,medium,2}{\eql@displaybreak@pre2}},%
4743   {{low,1}{\eql@displaybreak@pre1}},%
4744   {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
4745   {{default,inherit,-1}{\eql@displaybreak@pre@m@ne}}}}
4746 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4747   {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
4748   {{high,3}{\eql@displaybreak@post3}},%
4749   {{med,medium,2}{\eql@displaybreak@post2}},%
4750   {{low,1}{\eql@displaybreak@post1}},%
4751   {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
4752   {{default,inherit,-1}{\eql@displaybreak@post@m@ne}}}}
4753 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4754   \eql@decide@select{#3}{#2}{#1}{%
4755     {{full,4}{\eql@displaybreak@inter4}},%
4756     {{high,3}{\eql@displaybreak@inter3}},%
4757     {{med,medium,2}{\eql@displaybreak@inter2}},%
4758     {{low,1}{\eql@displaybreak@inter1}},%
4759     {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
4760 \eql@define@key{equations}{prepenalty}{%
4761   \eql@displaybreak@prepen@\numexpr#1\relax}
4762 \eql@define@key{equations}{postpenalty}{%
4763   \eql@displaybreak@postpen@\numexpr#1\relax}
4764 \eql@define@key{equations,setup}{interpenalty}{%
4765   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

4766 \eql@define@key{control}{vspace}[]{\eql@vspace@add{#1}}
4767 \eql@define@key{control}{vspace*}[]{\eql@vspace@addfixedbefore{#1}}
4768 \eql@define@key{control}{vspace!}[]{\eql@vspace@addfixedafter{#1}}
4769 \eql@define@key{control}{break}[4]{\eql@displaybreak@level{#1}}
4770 \eql@define@key{control}{penalty}[]{\eql@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4771 \eql@define@key\eql@keycat{displayheight}[strut]{%
4772   \eql@decide@select{#3}{#2}{#1}{%
4773     {\eql@decide@false{\let\eql@display@height\@undefined}},%
4774     {{strut}{\def\eql@display@height{\ht\eql@strutbox@}}},%
4775     {\relax{\def\eql@display@height{#1}}}}}}
4776 \eql@define@key\eql@keycat{displaydepth}[strut]{%
4777   \eql@decide@select{#3}{#2}{#1}{%
4778     {\eql@decide@false{\let\eql@display@depth\@undefined}},%
4779     {{strut}{\def\eql@display@depth{\dp\eql@strutbox@}}},%
4780     {\relax{\def\eql@display@depth{#1}}}}}}

```

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

```

4781 \eql@define@key{equations}{noskip}[]{%
4782   \eql@decide@abovebelow{#3}{#2}{#1}%

```

```

4783   {\def\eql@skip@force@above{5}}%
4784   {\def\eql@skip@force@below{5}}%
4785 \eql@define@key{equations}{short}[above]{%
4786   \eql@decide@abovebelow{#3}{#2}{#1}%
4787   {\def\eql@skip@force@above{1}}%
4788   {\def\eql@skip@force@below{1}}%
4789 \eql@define@key{equations}{long}[]{%
4790   \eql@decide@abovebelow{#3}{#2}{#1}%
4791   {\def\eql@skip@force@above{0}}%
4792   {\def\eql@skip@force@below{0}}%
4793 \eql@define@key{equations}{medskip}[]{%
4794   \eql@decide@abovebelow{#3}{#2}{#1}%
4795   {\def\eql@skip@force@above{6}}%
4796   {\def\eql@skip@force@below{6}}%
4797 \eql@define@key{equations}{par}[par]{%
4798   \eql@decide@select{#3}{#2}{#1}{%
4799     {{default},{\let\eql@skip@force@leave\@undefined}},%
4800     {{cont,hmode}{\let\eql@skip@force@leave\z@}},%
4801     {{par,vmode}{\let\eql@skip@force@leave\@ne
4802       \ifdefined\eql@skip@force@below\else
4803         \def\eql@skip@force@below{3}%
4804       \fi}},%
4805     {{top}{\let\eql@skip@force@leave\tw@
4806       \ifdefined\eql@skip@force@below\else
4807         \def\eql@skip@force@below{4}
4808       \fi}}}}

```

Specify vertical spacing explicitly:

```

4809 \eql@define@key{equations}{skip}{%
4810   \def\eql@skip@force@above{7}%
4811   \def\eql@skip@custom@above{#1}%
4812   \let\eql@skip@force@below\eql@skip@force@above
4813   \let\eql@skip@custom@below\eql@skip@custom@above}
4814 \eql@define@key{equations}{aboveskip}{%
4815   \def\eql@skip@force@above{7}%
4816   \def\eql@skip@custom@above{#1}}
4817 \eql@define@key{equations}{belowskip}{%
4818   \def\eql@skip@force@below{7}%
4819   \def\eql@skip@custom@below{#1}}
4820 \eql@define@key{equations}{abovespace}{%
4821   \advance\eql@abovespace@glueexpr#1\relax}
4822 \eql@define@key{equations}{belowspace}{%
4823   \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4824 \eql@define@key{intertext}{skip}{%
4825   \def\eql@skip@force@above{7}%
4826   \def\eql@skip@custom@above{#1}%
4827   \let\eql@skip@force@below\eql@skip@force@above
4828   \let\eql@skip@custom@below\eql@skip@custom@above}
4829 \eql@define@key{intertext}{aboveskip}{%
4830   \def\eql@skip@force@below{7}%
4831   \def\eql@skip@custom@below{#1}}
4832 \eql@define@key{intertext}{belowskip}{%
4833   \def\eql@skip@force@above{7}%
4834   \def\eql@skip@custom@above{#1}}
4835 \eql@define@key{intertext}{noskip}[]{%
4836   \eql@decide@abovebelow{#3}{#2}{#1}%

```

```

4837     {\def\eql@skip@force@below{5}}%
4838     {\def\eql@skip@force@above{5}}%
4839 \eql@define@key{intertext}{short}[] {%
4840   \eql@decide@abovebelow{#3}{#2}{#1}%
4841   {\def\eql@skip@force@below{1}}%
4842   {\def\eql@skip@force@above{1}}%
4843 \eql@define@key{intertext}{long}[] {%
4844   \eql@decide@abovebelow{#3}{#2}{#1}%
4845   {\def\eql@skip@force@below{0}}%
4846   {\def\eql@skip@force@above{0}}%
4847 \eql@define@key{intertext}{medskip}[] {%
4848   \eql@decide@abovebelow{#3}{#2}{#1}%
4849   {\def\eql@skip@force@below{6}}%
4850   {\def\eql@skip@force@above{6}}%

```

Configure general vertical spacing behaviour for various situations:

```

4851 \eql@define@key{setup}{skip,longskip}{%
4852   \abovedisplayskip\glueexpr#1\relax
4853   \belowdisplayskip\abovedisplayskip
4854   \def\eql@skip@long@above{#1}%
4855   \let\eql@skip@long@below\eql@skip@long@above}
4856 \eql@define@key{setup}{aboveskip,abovelongskip}{%
4857   \abovedisplayskip\glueexpr#1\relax
4858   \def\eql@skip@long@above{#1}}
4859 \eql@define@key{setup}{belowskip,belowlongskip}{%
4860   \belowdisplayskip\glueexpr#1\relax
4861   \def\eql@skip@long@below{#1}}
4862 \eql@define@key{setup}{aboveshortskip}{%
4863   \abovedisplayshortskip\glueexpr#1\relax
4864   \def\eql@skip@short@above{#1}}
4865 \eql@define@key{setup}{belowshortskip}{%
4866   \belowdisplayshortskip\glueexpr#1\relax
4867   \def\eql@skip@short@below{#1}}
4868 \eql@define@key{setup}{tagskip}{%
4869   \def\eql@skip@tag@above{#1}%
4870   \let\eql@skip@tag@below\eql@skip@tag@above}
4871 \eql@define@key{setup}{abovetagskip}{%
4872   \def\eql@skip@tag@above{#1}}
4873 \eql@define@key{setup}{belowtagskip}{%
4874   \def\eql@skip@tag@below{#1}}
4875 \eql@define@key{setup}{medskip}{%
4876   \def\eql@skip@med@above{#1}%
4877   \let\eql@skip@med@below\eql@skip@med@above}
4878 \eql@define@key{setup}{abovemedskip}{%
4879   \def\eql@skip@med@above{#1}}
4880 \eql@define@key{setup}{belowmedskip}{%
4881   \def\eql@skip@med@below{#1}}
4882 \eql@define@key{setup}{abovetopskip}{%
4883   \def\eql@skip@top@above{#1}}
4884 \eql@define@key{setup}{belowtopskip}{%
4885   \def\eql@skip@top@below{#1}}
4886 \eql@define@key{setup}{aboveparskip}{%
4887   \def\eql@skip@par@above{#1}}
4888 \eql@define@key{setup}{belowparskip}{%
4889   \def\eql@skip@par@below{#1}}
4890 \eql@define@key{setup}{abovecontskip}{%
4891   \eql@decide@select{#3}{#2}{#1}{%
4892     {\hide}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%

```



```

4893     {\relax{\def\eq@skip@cont@above{#1}}}}
4894 \eq@define@key{setup}{belowcontskip}{%
4895   \def\eq@skip@cont@below{#1}}
4896 \eq@define@key{setup}{shortmode}{%
4897   \eq@decide@select{#3}{#2}{#1}{%
4898     {{off,never,no}{\def\eq@skip@mode@short{0}}},%
4899     {{above,neverbelow,notbelow,belowoff}{\def\eq@skip@mode@short{1}}},%
4900     {{belowone,belowsingle}{\def\eq@skip@mode@short{2}}},%
4901     {{belowall,always,on}{\def\eq@skip@mode@short{3}}}}
4902 \eq@define@key{setup}{abovecontmode}{%
4903   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
4904 \eq@define@key{setup}{belowcontmode}{%
4905   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
4906 \eq@define@key{setup}{aboveparmode}{%
4907   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@above}
4908 \eq@define@key{setup}{belowparmode}{%
4909   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@below}
4910 \eq@define@key{setup}{abovetopmode}{%
4911   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@above}
4912 \eq@define@key{setup}{belowtopmode}{%
4913   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@below}

```

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

```

4914 \def\eq@keycat{equations,subequations}
4915 \eq@define@key\eq@keycat{label}{\eq@tags@addblock@label{#1}}
4916 \eq@define@key\eq@keycat{labelname}{\eq@tags@addblock@name{#1}}
4917 \eq@define@key\eq@keycat{tag}{\eq@tags@addblock@tag{#1}}
4918 \eq@define@key\eq@keycat{tag*}{%
4919   \eq@tags@addblock@tagform@off\eq@tags@addblock@tag{#1}}
4920 \eq@define@key\eq@keycat{taglabel}{\eq@tags@addblock@ref{#1}}

```

TODO: describe

```

4921 \eq@define@key{control}{label}{\eq@tags@add@label{#1}}
4922 \eq@define@key{control}{labelname}{\eq@tags@add@name{#1}}
4923 \eq@define@key{control}{tag}{\eq@tags@add@tag{#1}}
4924 \eq@define@key{control}{tag*}{\eq@tags@add@tagform@off\eq@tags@add@tag{#1}}
4925 \eq@define@key{control}{taglabel}{\eq@tags@add@ref{#1}}
4926 \eq@define@key{control}{shifttag}{\eq@tags@add@raiseshift{#1}}
4927 \eq@define@key{control}{smashtag}{\eq@tags@add@raisesmash{#1}}
4928 \eq@define@key{control}{pushtag}{\eq@tags@add@forceraise}

```

TODO: describe

```

4929 \eq@define@key{setup}{labelname}{\protected@edef\eq@tags@name@generic{#1}}
4930 \eq@define@key{setup}{autolabel}[true]{%
4931   \eq@decide@bool{#3}{#2}{#1}\eq@tags@autolabel}
4932 \eq@define@key{setup}{autotag}[true]{%
4933   \eq@decide@bool{#3}{#2}{#1}\eq@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

4934 \def\eq@keycat{equations,setup}
4935 \eq@define@key\eq@keycat{tagmargin}[auto]{%
4936   \eq@decide@select{#3}{#2}{#1}{%
4937     {{auto,\eq@decide@false}{\let\eq@tagmargin@val\undefined}},%
4938     {\relax{\def\eq@tagmargin@val{#1}}}}}
4939 \eq@define@key\eq@keycat{tagmargin*}{%

```

```

4940 \settowidth\dimen@{#1}\edef\eql@tagmargin@val{\the\dimen@}
4941 \eql@define@key\eql@keycat{tagmarginratio}{%
4942 \eql@tagmargin@ratio@{\dimexpr#1pt\relax}
4943 \eql@define@key\eql@keycat{tagmarginthreshold}{%
4944 \def\eql@tagmargin@threshold{#1}}
4945 \eql@define@key\eql@keycat{mintagsep}{\def\eql@tagsepmin@val{#1}}
4946 \eql@define@key\eql@keycat{mintagwidth}{%
4947 \settowidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}
4948 \eql@define@key\eql@keycat{mintagwidth*}{\settowidth\eql@tagwidthmin@{#1}}
4949 \eql@define@key\eql@keycat{tagsnap}{%
4950 \eql@decide@select{#3}{#2}{#1}{%
4951 {\eql@decide@false{\let\eql@tagpos@snap@z@},%
4952 {\relax{\def\eql@tagpos@snap{#1}}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

4953 \def\eql@keycat{equations,setup}
4954 \eql@define@key\eql@keycat{tagbox,taglayout}{%
4955 \eql@tags@taglayout@set{#1}}
4956 \eql@define@key\eql@keycat{tagbox*,taglayout*}{%
4957 \eql@tags@taglayout@set@direct{#1}}
4958 \eql@define@key\eql@keycat{tagform}{%
4959 \eql@tags@tagform@set{#1}}
4960 \eql@define@key\eql@keycat{tagform*}{%
4961 \eql@tags@tagform@set@direct{#1}}
4962 \eql@define@key\eql@keycat{subeqtemplate}{%
4963 \def\eql@subequations@template####1####2{#1}%
4964 \eql@append\eql@subequations@template{\theparentequation{equation}}}}

4965 \eql@define@key{control}{tagbox,taglayout}{%
4966 \global\eql@append\eql@tags@container{\eql@tags@taglayout@set{#1}}}
4967 \eql@define@key{control}{tagbox*,taglayout*}{%
4968 \global\eql@append\eql@tags@container{\eql@tags@taglayout@set@direct{#1}}}
4969 \eql@define@key{control}{tagform}{%
4970 \global\eql@append\eql@tags@container{\eql@tags@tagform@set{#1}}}
4971 \eql@define@key{control}{tagform*}[{####1}]{%
4972 \global\eql@append\eql@tags@container{\eql@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

4973 \def\eql@keycat{equations,setup}
4974 \eql@define@key\eql@keycat{numberline,number,num,numline,n}[all]{%
4975 \eql@decide@select{#3}{#2}{#1}{%
4976 {\eql@decide@false,0,*}{\let\eql@numbering@active\eql@false}},%
4977 {\eql@decide@true,!}{\let\eql@numbering@active\eql@true}},%
4978 {\none,n,-}{\let\eql@numbering@mode\eql@numbering@mode@multi
4979 \let\eql@numbering@active\eql@false}},%
4980 {\single,1}{\let\eql@numbering@mode\eql@numbering@mode@single
4981 \let\eql@numbering@active\eql@true}},%
4982 {\multi,@}{\let\eql@numbering@mode\eql@numbering@mode@multi
4983 \let\eql@numbering@active\eql@true}},%
4984 {\relax{\eql@numbering@set{#1}}}}}
4985 \eql@define@key\eql@keycat{nonumber,nn,*}[]{%
4986 \let\eql@numbering@active\eql@false}
4987 \eql@define@key\eql@keycat{donumber,dn,!}[]{%
4988 \let\eql@numbering@active\eql@true}
4989 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
4990 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}

```

```

4991 \eqld@define@key\eqld@keycat{tags,eqno}{%
4992   \eqld@decide@select{#3}{#2}{#1}{%
4993     {{right,r}}{\let\eqld@tagsleft\eqld@false}},%
4994     {{left,l}}{\let\eqld@tagsleft\eqld@true}}}%
4995 \eqld@define@key\eqld@keycat{evadetag,avoidtag}[true]{%
4996   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}
4997 \eqld@define@key\eqld@keycat{tagbetween}[true]{%
4998   \eqld@decide@bool{#3}{#2}{#1}\eqld@tagpos@doconvert}

```

TODO: describe

```

4999 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
5000 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
5001 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
5002 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

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

```

5003 \def\eqld@keycat{equations,setup}
5004 \eqld@define@key\eqld@keycat{layout}{\eqld@decide@select{#3}{#2}{#1}{%
5005   {{center,c}}{\let\eqld@layoutleft\eqld@false}},%
5006   {{left,l}}{\let\eqld@layoutleft\eqld@true}}}%
5007 \eqld@define@key\eqld@keycat{center}[]{\let\eqld@layoutleft\eqld@false}
5008 \eqld@define@key\eqld@keycat{flushleft,left}[]{\let\eqld@layoutleft\eqld@true}
5009 \eqld@define@key\eqld@keycat{leftmargin}{\def\eqld@layoutleftmargin{#1}}
5010 \eqld@define@key\eqld@keycat{leftmargin*}{%
5011   \settowidth\dimen@{#1}\edef\eqld@layoutleftmargin{\the\dimen@}}
5012 \eqld@define@key\eqld@keycat{minleftmargin}{%
5013   \def\eqld@layoutleftmarginmin{#1}}
5014 \eqld@define@key\eqld@keycat{maxleftmargin}{%
5015   \eqld@decide@select{#3}{#2}{#1}{%
5016     {\eqld@decide@false{\def\eqld@layoutleftmarginmax{.5\maxdimen}}},%
5017     {\relax{\def\eqld@layoutleftmarginmax{#1}}}}}%
5018 \def\eqld@keycat{equations,equationsbox}
5019 \eqld@define@key\eqld@keycat{margin}{%
5020   \def\eqld@display@marginleft{#1}\def\eqld@display@marginright{#1}}
5021 \eqld@define@key\eqld@keycat{marginleft}{\def\eqld@display@marginleft{#1}}
5022 \eqld@define@key\eqld@keycat{marginright}{\def\eqld@display@marginright{#1}}
5023 \eqld@define@key\eqld@keycat{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

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

```

5024 \def\eqld@keycat{equations,setup}
5025 \eqld@define@key\eqld@keycat{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5026   {{max,full,4}}{\eqld@alignbadness@inf@bad}},%
5027   {{high,3}}{\eqld@alignbadness@54\relax}},%
5028   {{med,medium,2}}{\eqld@alignbadness@18\relax}},%
5029   {{low,1}}{\eqld@alignbadness@6\relax}},%
5030   {{0,\eqld@decide@false}}{\eqld@alignbadness@z@}}}%
5031 \eqld@define@key\eqld@keycat{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5032   {{max,full,4}}{\eqld@tagbadness@inf@bad}},%
5033   {{high,3}}{\eqld@tagbadness@54\relax}},%
5034   {{med,medium,2}}{\eqld@tagbadness@18\relax}},%
5035   {{low,1}}{\eqld@tagbadness@6\relax}},%
5036   {{0,\eqld@decide@false}}{\eqld@tagbadness@z@}}}%

```

```

5037 \eqld@define@key\eqld@keycat{alignbadness}{\eqld@alignbadness@numexpr#1\relax}
5038 \eqld@define@key\eqld@keycat{tagbadness}{\eqld@tagbadness@numexpr#1\relax}
5039 \eqld@define@key\eqld@keycat{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5040   {0,\eqld@decide@false}{\def\eqld@colsepmin@val{0pt}}},%
5041   {\relax{\def\eqld@colsepmin@val{#1}}}}}%
5042 \eqld@define@key\eqld@keycat{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5043   {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
5044   {\relax{\def\eqld@colsepmax@val{#1}}}}}%
5045 \eqld@define@key\eqld@keycat{fulllength}[true]{%
5046   \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

5047 \eqld@define@key\eqld@keycat{linesep}{\eqld@decide@select{#3}{#2}{#1}{%
5048   {0,\eqld@decide@false}{\def\eqld@break@line@sep{0pt}}},%
5049   {\relax{\def\eqld@break@line@sep{#1}}}}}%
5050 \eqld@define@key\eqld@keycat{linesep*}{\eqld@decide@select{#3}{#2}{#1}{%
5051   {0,\eqld@decide@false}{\def\eqld@break@line@shortsep{0pt}}},%
5052   {\relax{\def\eqld@break@line@shortsep{#1}}}}}%
5053 \eqld@define@key{equationsbox,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5054   {0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
5055   {\relax{\def\eqld@break@col@sep{#1}}}}}%
5056 \let\eqld@box@colsep\eqld@break@col@sep
5057 \eqld@define@key{equations}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5058   {0,\eqld@decide@false}{\def\eqld@break@col@sep{0pt}}},%
5059   {\relax{\def\eqld@break@col@sep{#1}}}}}%
5060 \let\eqld@colsepmin@val\eqld@box@colsep
5061 \let\eqld@colsepmax@val\eqld@box@colsep
5062 \let\eqld@box@colsep\eqld@break@col@sep
5063 \eqld@define@key\eqld@keycat{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
5064   {0,\eqld@decide@false}{\def\eqld@break@col@shortsep{0pt}}},%
5065   {\relax{\def\eqld@break@col@shortsep{#1}}}}}%

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5066 \def\eqld@keycat{equations,equationsbox,setup}
5067 \eqld@define@key\eqld@keycat{shape}[default]{\eqld@shape@set{#1}}
5068 \eqld@define@key\eqld@keycat{padding,pad}[indent]{%
5069   \eqld@decide@select{#3}{#2}{#1}{%
5070     {\max}{\let\eqld@paddingleft@val\undefined}},%
5071     {{indent}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
5072     {{0,\eqld@decide@false}{\def\eqld@paddingleft@val{0pt}}},%
5073     {\relax{\def\eqld@paddingleft@val{#1}}}}}%
5074 \let\eqld@paddingright@val\eqld@paddingleft@val
5075 \eqld@define@key\eqld@keycat{padleft}[indent]{%
5076   \eqld@decide@select{#3}{#2}{#1}{%
5077     {\max}{\let\eqld@paddingleft@val\undefined}},%
5078     {{indent}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%
5079     {{0,\eqld@decide@false}{\def\eqld@paddingleft@val{0pt}}},%
5080     {\relax{\def\eqld@paddingleft@val{#1}}}}}%
5081 \eqld@define@key\eqld@keycat{padright}[indent]{%
5082   \eqld@decide@select{#3}{#2}{#1}{%
5083     {\max}{\let\eqld@paddingright@val\undefined}},%
5084     {{indent}{\def\eqld@paddingright@val{\eqld@indent@val}}},%
5085     {{0,\eqld@decide@false}{\def\eqld@paddingright@val{0pt}}},%
5086     {\relax{\def\eqld@paddingright@val{#1}}}}}%
5087 \eqld@define@key\eqld@keycat{indent}[2em]{%
5088   \def\eqld@indent@val{#1}}

```

TODO: describe

```

5089 \eql@define@key{control}{align}[]{%
5090   \eql@decide@select{#3}{#2}{#1}{%
5091     {l,left}{\global\eql@append\eql@cell@container{\eql@shape@pos@z@}}},%
5092     {c,center}{\global\eql@append\eql@cell@container{\eql@shape@pos@ne@}}},%
5093     {r,right}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw@}}}}%
5094 \eql@define@key{control}{shift,shiftto}[]{%
5095   \eql@decide@select{#3}{#2}{#1}{%
5096     {*,indent}{\eql@shape@alignamount@set{\eql@indent@}}},%
5097     {!,outdent}{\eql@shape@alignamount@set{-\eql@indent@}}},%
5098     {relax{\eql@shape@alignamount@set{#1}}}}%
5099 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

5100 \def\eql@keycat{equations,equationsbox,setup}
5101 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
5102 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
5103 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
5104 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
5105 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
5106 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}{%
5107   {ampeq,amprel,eafter,beforerel}\eql@class@ampeq},%
5108   {eqamp,relamp,eqbefore,afterrel}\eql@class@eqamp}}}

```

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

```

5109 \def\eql@punct@all#1#2#3#4#5\eql@punct@end{%
5110   \def\eql@tmp{#4}\def\eql@tmpa{1}%
5111   \ifx\eql@tmp\eql@tmpa
5112     \ifnum#5=1111\relax
5113       \eql@punct@set\eql@punct@col{#1}%
5114       \eql@punct@set\eql@punct@line{#2}%
5115       \eql@punct@set\eql@punct@main{#3}%
5116     \else\ifnum#5=111\relax
5117       \eql@punct@set\eql@punct@line{#1}%
5118       \eql@punct@set\eql@punct@main{#2}%
5119     \else\ifnum#5=11\relax
5120       \eql@punct@set\eql@punct@main{#1}%
5121     \else
5122       \let\eql@punct@col\@empty
5123       \let\eql@punct@line\@empty
5124       \let\eql@punct@main\@empty
5125     \fi\fi\fi
5126   \else
5127     \eql@error{Too many arguments to punctall}%
5128   \fi
5129 }

```

TODO: describe

```

5130 \def\eql@keycat{equations,equationsbox,setup}
5131 \eql@define@key\eql@keycat{punctsep}{[,]\}{\def\eql@punct@sep{#1}}
5132 \eql@define@key\eql@keycat{punct}{.}{\eql@punct@set\eql@punct@main{#1}}
5133 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
5134 \eql@define@key\eql@keycat{punctline}{,}{\eql@punct@set\eql@punct@line{#1}}
5135 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
5136 \eql@define@key\eql@keycat{punctcol}{,}{\eql@punct@set\eql@punct@col{#1}}
5137 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}

```

```

5138 \eqld@define@key\eqld@keycat{punctall}[,,.]{\eqld@punct@all#111111\eqld@punct@end}

5139 \eqld@define@key{control}{punctsep}[,,.]{\def\eqld@punct@sep{#1}}
5140 \eqld@define@key{control}{punct}[.]{\eqld@punct@set\eqld@punct@block{#1}%
5141   \eqld@punct@set\eqld@punct@line{#1}\eqld@punct@set\eqld@punct@col{#1}}
5142 \eqld@define@key{control}{punct*}[]{\let\eqld@punct@block\relax}
5143 \eqld@define@key{control}{punctapply}[]{\eqld@punct@apply@block}

```

Frames. **TODO:** describe

```

5144 \eqld@define@key{equationsbox}{frame}[\fbox]{%
5145   \def\eqld@box@frame{#1}%
5146   \ifx\eqld@box@frame\empty\let\eqld@box@frame\@firstofone\fi}
5147 \eqld@define@key{equationsbox}{wrap}[]{\eqld@box@wrap#1}

```

TODO: describe

```

5148 \eqld@define@key{control}{framecell}[\fbox]{%
5149   \global\eqld@append\eqld@cell@container{\def\eqld@frame@cmd{#1}}
5150 \eqld@define@key{control}{frametag}[\fbox]{%
5151   \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

```

5152 \eqld@define@key{equations,equationsbox}{alt}{}

```

Injectsions.

```

5153 \eqld@define@key{control}{inject}{%
5154   \global\eqld@append\eqld@interline@container{%
5155     \eqld@append\eqld@display@injectbefore{#1}}
5156 \eqld@define@key{control}{inject*}{%
5157   \global\eqld@append\eqld@interline@container{%
5158     \eqld@append\eqld@display@injectafter{#1}}
5159 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
5160 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
5161 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
5162 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}

```

TODO: describe

```

5163 \eqld@define@key{markline}{pos}{%
5164   \eqld@decide@select{#3}{#2}{#1}{%
5165     {{below,push}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5166     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5167     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}
5168 \eqld@define@key{markline}{below,push}[]{%
5169   \let\eqld@markline@pos\eqld@markline@pos@below}
5170 \eqld@define@key{markline}{baseline}[]{%
5171   \let\eqld@markline@pos\eqld@markline@pos@baseline}
5172 \eqld@define@key{markline}{bottom}[]{%
5173   \let\eqld@markline@pos\eqld@markline@pos@bottom}
5174 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
5175 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}
5176 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
5177 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}
5178 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
5179 \eqld@define@key{setup}{markpos}{%

```

```

5180 \eq@decide@select{#3}{#2}{#1}{%
5181   {{below}}{\let\eq@markline@pos\eq@markline@pos@below}},%
5182   {{baseline}}{\let\eq@markline@pos\eq@markline@pos@baseline}},%
5183   {{bottom}}{\let\eq@markline@pos\eq@markline@pos@bottom}}}}

```

Global Switches. Set global switches:

```

5184 \let\eq@multi@linesfallback\eq@false
5185 \let\eq@scan@par\eq@false
5186 \let\eq@single@cr@mode\eq@false
5187 \let\eq@ampproof@active\eq@false

5188 \eq@define@key{equations,setup}{linesfallback}[true]{%
5189   \eq@decide@select{#3}{#2}{#1}{%
5190     {\eq@decide@false{\let\eq@multi@linesfallback\eq@false}},%
5191     {{reuse,lean}}{\let\eq@multi@linesfallback\z@}},%
5192     {{measure,full,\eq@decide@true}}{\let\eq@multi@linesfallback\eq@true}}}}
5193 \eq@define@key{setup}{ampproof}[true]{%
5194   \eq@decide@bool{#3}{#2}{#1}\eq@ampproof@active}
5195 \eq@define@key{equations,setup}{equationcr}{%
5196   \eq@decide@select{#3}{#2}{#1}{%
5197     {\eq@decide@false{\let\eq@single@cr@mode\eq@false}},%
5198     {{\eq@decide@true,break}}{\let\eq@single@cr@mode\eq@break@line}},%
5199     {{error,verbose}}{\let\eq@single@cr@mode\eq@single@cr@error}}}}
5200 \eq@define@key{setup}{modifierwarning}[true]{%
5201   \eq@decide@select{#3}{#2}{#1}{%
5202     {\eq@decide@false{\let\eq@parseopt@warn\@empty}},%
5203     {\eq@decide@true{\let\eq@parseopt@warn\eq@warn@parseopt}},%
5204     {{verbose,+}}{\let\eq@parseopt@warn\eq@warn@parseopt@verbose}}}}
5205 \let\eq@parseopt@warn\eq@warn@parseopt
5206 \eq@define@key{equations,setup}{rescan}[true]{%
5207   \eq@decide@if{#3}{#2}{#1}%
5208   {\let\eq@scan@body\eq@scan@body@rescan}%
5209   {\let\eq@scan@body\eq@scan@body@dump}}
5210 \eq@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5211   \eq@decide@bool{#3}{#2}{#1}\eq@scan@par}
5212 \eq@define@key{setup}{defaults}{%
5213   \eq@decide@select{#3}{#2}{#1}{%
5214     {{classic}}{\eq@defaults@classic}},%
5215     {{eqnlines}}{\eq@defaults@eqnlines}}}}
5216 \eq@define@key{equations,equationsbox,setup}{verbose}[true]{%
5217   \eq@decide@if{#3}{#2}{#1}\eq@verbose@on\eq@verbose@off}

```

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

```

5218 \let\eq@provide@opt@env\tw@
5219 \let\eq@provide@opt@amsmathends\eq@true
5220 \let\eq@provide@opt@backup\eq@false
5221 \let\eq@provide@opt@ang\eq@true
5222 \let\eq@provide@opt@eqref\eq@true

5223 \eq@define@key{setup}{amsmathends}[true]{%
5224   \eq@error@packageoption{#2}%
5225   \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@amsmathends}
5226 \eq@define@key{setup}{backup}[true]{%
5227   \eq@error@packageoption{#2}%
5228   \eq@decide@bool{#3}{#2}{#1}\eq@provide@opt@backup}
5229 \eq@define@key{setup}{env}[equation]{%

```



```

5230 \eqLError@packageoption{#2}%
5231 \eql@decide@select{#3}{#2}{#1}{%
5232   {\none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5233   {\equation,latex}{\let\eql@provide@opt@env\@ne}},%
5234   {\amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}%
5235 \eql@define@key{setup}{ang}[true]{%
5236   \eqLError@packageoption{#2}%
5237   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5238 \eql@define@key{setup}{eqref}[true]{%
5239   \eqLError@packageoption{#2}%
5240   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Shortcut Options. **TODO:** describe

```

5241 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5242 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5243 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5244 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5245 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5246 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5247 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5248 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5249 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
5250 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
5251 \def\eql@parseopt@punctoff#1{\eqnaddopt{punct={}}\eql@parseopt@peek}
5252 \def\eql@parseopt@punctall#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}

```

TODO: describe

```

5253 \def\eql@parseopt@enddot#1{%
5254   \eql@punct@set\eql@punct@block.\eql@parseopt@peek}
5255 \def\eql@parseopt@endcomma#1{%
5256   \eql@punct@set\eql@punct@block,\eql@parseopt@peek}
5257 \def\eql@parseopt@endoff#1{%
5258   \eql@punct@set\eql@punct@block~\eql@parseopt@peek}
5259 \def\eql@parseopt@endpar#1#2{%
5260   \eql@punct@set\eql@punct@block{#2}\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:

```

5261 \def\eql@defaults@classic{%
5262   \eqnlineset{numberline=all}%
5263   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5264   \eqnlineset{maxcolsep=off}%
5265   \eqnlineset{spread={\jot}}%
5266   \eqnlineset{tagmargin}%
5267   \eqnlineset{tagmarginratio=1}%
5268   \eqnlineset{tagmarginthreshold=0.5}%
5269   \eqnlineset{leftmargin={\leftmargini}}%
5270   \eqnlineset{padding=max}%
5271   \eqnlineset{evadetag=off}%

```



```

5272 \eqnlineset{displayheight=off}%
5273 \eqnlineset{displaydepth=off}%
5274 \eqnlineset{shortmode=belowsingle}%
5275 \eqnlineset{abovecontmode=short}%
5276 \eqnlineset{belowcontmode=short}%
5277 \eqnlineset{aboveparmode=long}%
5278 \eqnlineset{belowparmode=long}%
5279 \eqnlineset{abovetopmode=long}%
5280 \eqnlineset{belowtopmode=long}%
5281 \eqnlineset{abovelongskip={\abovedisplayskip}}%
5282 \eqnlineset{belowlongskip={\belowdisplayskip}}%
5283 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5284 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5285 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5286 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5287 \eqnlineset{abovecontskip=0pt}%
5288 \eqnlineset{belowcontskip=0pt}%
5289 \eqnlineset{aboveparskip=0pt}%
5290 \eqnlineset{belowparskip=0pt}%
5291 \eqnlineset{abovetopskip=0pt}%
5292 \eqnlineset{belowtopskip=0pt}%
5293 \eqnlineset{abovetagskip=0pt}%
5294 \eqnlineset{belowtagskip=0pt}%
5295 \eqnlineset{equationcr=off}%
5296 \eqnlineset{linesfallback=false}%
5297 }

```

values based on 10pt vs 12pt

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

```

5298 \def\eq1@defaults@eqnlines{%
5299   \eqnlineset{numberline=all}%
5300   \eqnlineset{mintagsep=.5em}%
5301   \eqnlineset{maxcolsep=2em}%
5302   \eqnlineset{spread={0.2\normalbaselineskip}}%
5303   \eqnlineset{tagmargin}%
5304   \eqnlineset{tagmarginratio=.334}%
5305   \eqnlineset{tagmarginthreshold=0.5}%
5306   \eqnlineset{leftmargin={\leftmargini}}%
5307   \eqnlineset{padding=0pt}%
5308   \eqnlineset{evadetag}%
5309   \eqnlineset{displayheight=strut}%
5310   \eqnlineset{displaydepth=strut}%
5311   \eqnlineset{shortmode=above}%
5312   \eqnlineset{abovecontmode=noskip}%
5313   \eqnlineset{belowcontmode=long}%
5314   \eqnlineset{aboveparmode=long}%
5315   \eqnlineset{belowparmode=long}%
5316   \eqnlineset{abovetopmode=noskip}%
5317   \eqnlineset{belowtopmode=long}%
5318   \eqnlineset{longskip={0.75\normalbaselineskip
5319     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5320   \eqnlineset{aboveshortskip={0.0\normalbaselineskip

```

```

5321   plus 0.25\normalbaselineskip}}%
5322 \eqnlineset{belowshortskip={0.0\normalbaselineskip
5323   plus 0.25\normalbaselineskip}}%
5324 \eqnlineset{medskip={0.4\normalbaselineskip
5325   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5326 \eqnlineset{abovecontskip=0pt}%
5327 \eqnlineset{belowcontskip=0pt}%
5328 \eqnlineset{aboveparskip=0pt}%
5329 \eqnlineset{belowparskip=0pt}%
5330 \eqnlineset{abovetopskip=0pt}%
5331 \eqnlineset{belowtopskip=0pt}%
5332 \eqnlineset{abovetagskip={0.25\normalbaselineskip
5333   minus 0.25\normalbaselineskip}}%
5334 \eqnlineset{belowtagskip={0.25\normalbaselineskip
5335   minus 0.25\normalbaselineskip}}%
5336 \eqnlineset{equationcr=break}%
5337 \eqnlineset{linesfallback=true}%
5338 }

```

16.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of `LATEX` and `amsmath` carefully.

Tools.

```

\eql@provide@movecmd We introduce a couple of tools to rename and undefine commands and environments:
\eql@provide@moveenv
@provide@undefinecmd
@provide@undefineenv
5339 \def\eql@provide@movecmd#1#2{%
5340   \eql@letcs{#1}\expandafter}\csname#2\endcsname
5341 }
5342 \def\eql@provide@moveenv#1#2{%
5343   \eql@provide@movecmd{#1}{#2}%
5344   \ifcsname end#2\endcsname
5345     \eql@provide@movecmd{end#1}{end#2}%
5346   \fi
5347 }
5348 \def\eql@provide@undefinecmd#1{%
5349   \eql@letcs{#1}\@undefined
5350 }
5351 \def\eql@provide@undefineenv#1{%
5352   \eql@provide@undefinecmd{#1}%
5353   \eql@provide@undefinecmd{end#1}%
5354 }

```

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:

```

```

5355 \def\eql@amsmath@endfix#1#2{%
5356   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
5357   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
5358     \eql@provide@movecmd{end#1}{end#2}%
5359   \fi
5360 }

```

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

```

5361 \def\eql@amsmath@fixends{%
5362   \eql@amsmath@after{%
5363     \eql@amsmath@endfix{flalign}{align}%
5364     \eql@amsmath@endfix{alignat}{align}%
5365     \eql@amsmath@endfix{xalignat}{align}%
5366     \eql@amsmath@endfix{xxalignat}{align}%
5367     \eql@amsmath@endfix{gather*}{gather}%
5368     \eql@amsmath@endfix{multline*}{multline}%
5369     \eql@amsmath@endfix{align*}{align}%
5370     \eql@amsmath@endfix{flalign*}{align}%
5371     \eql@amsmath@endfix{alignat*}{align}%
5372     \eql@amsmath@endfix{xalignat*}{align}%
5373     \eql@amsmath@endfix{gathered}{aligned}%
5374     \eql@amsmath@endfix{alignedat}{aligned}%
5375   }
5376 }

```

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

```

5377 \def\eql@provide@backup@amsenv#1{%
5378   \eql@amsmath@after{%
5379     \eql@provide@moveenv{ams#1}{#1}%
5380     \eql@tagging@register@luamml{ams#1}%
5381     \eql@markline@amsthm@move{ams#1}{#1}%
5382   }%
5383 }

```

`\provide@backup@amssub` **TODO:** describe

```

5384 \def\eql@provide@backup@amssub#1{%
5385   \eql@amsmath@after{%
5386     \eql@provide@moveenv{ams#1}{#1}%
5387   }%
5388 }

```

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

```

5389 \def\eql@provide@backup@eqref{%
5390   \eql@amsmath@after{%
5391     \eql@provide@movecmd{amseqref}{eqref}%
5392   }%
5393 }

```

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

```

5394 \def\eql@provide@backup@multlined{%
5395   \AddToHook{package/mathtools/after}{%
5396     \eql@provide@moveenv{amsmultlined}{multlined}}%
5397 }

```

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:

```

5398 \def\eql@provide@backup@equation{%
5399   \eql@amsmath@after{%
5400     \eql@provide@moveenv{amsequeation}{equation}%
5401     \eql@provide@moveenv{amsequeation*}{equation*}%
5402     \eql@tagging@register@env{amsequeation}%
5403     \eql@tagging@register@env{amsequeation*}%
5404     \eql@tagging@register@luamml{amsequeation}%
5405     \eql@tagging@register@luamml{amsequeation*}%
5406     \eql@markline@amsthm@move{amsequeation}{equation}%
5407     \eql@markline@amsthm@move{amsequeation*}{equation*}%
5408   }%
5409   \AddToHook{package/hyperref/after}{%
5410     \@ifpackageloaded{amsmath}{}%
5411     \eql@provide@moveenv{hyperrefequation}{equation}%
5412     \eql@tagging@register@env{hyperrefequation}%
5413     \eql@tagging@register@luamml{hyperrefequation}%
5414     \eql@markline@amsthm@move{hyperrefequation}{equation}%
5415   }%
5416 }%
5417 \@ifpackageloaded{amsmath}{}%
5418   \@ifpackageloaded{hyperref}{%
5419     \let\latexequation\H@equation
5420     \let\endlatexequation\H@endequation
5421   }{\eql@provide@moveenv{latexequation}{equation}}%
5422   \eql@tagging@register@env{latexequation}%
5423   \eql@tagging@register@luamml{latexequation}%
5424   \eql@markline@amsthm@move{latexequation}{equation}%
5425 }%
5426 }

```

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

```

5427 \def\eql@provide@backup@displaymath{%
5428   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5429   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
5430 }

```

@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:

```

5431 \def\eql@provide@backup@subequations{%
5432   \eql@amsmath@after{%
5433     \eql@provide@moveenv{amssubequations}{subequations}%

```

```

5434 }%
5435 \AddToHook{package/hyperref/after}{%
5436   \AddToHook{cmd/amssubequations/before}{%
5437     {%
5438       \stepcounter{equation}%
5439       \protected@edef\theHparentequation{\theHequation}%
5440       \addtocounter{equation}{-1}%
5441     }%
5442   \AddToHook{cmd/amssubequations/after}{%
5443     {%
5444       \def\theHequation{\theHparentequation\alph{equation}}%
5445       \ignorespaces
5446     }%
5447   }%
5448 }

```

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

```

5449 \def\eql@provide@backup{%
5450   \eql@provide@backup@eqref
5451   \eql@provide@backup@equation
5452   \eql@provide@backup@displaymath
5453   \eql@provide@backup@amsenv{gather}%
5454   \eql@provide@backup@amsenv{multline}%
5455   \eql@provide@backup@amsenv{align}%
5456   \eql@provide@backup@amsenv{flalign}%
5457   \eql@provide@backup@amsenv{alignat}%
5458   \eql@provide@backup@amsenv{xalignat}%
5459   \eql@provide@backup@amsenv{xxalignat}%
5460   \eql@provide@backup@amsenv{gather*}%
5461   \eql@provide@backup@amsenv{multline*}%
5462   \eql@provide@backup@amsenv{align*}%
5463   \eql@provide@backup@amsenv{flalign*}%
5464   \eql@provide@backup@amsenv{alignat*}%
5465   \eql@provide@backup@amsenv{xalignat*}%
5466   \eql@provide@backup@amssub{aligned}%
5467   \eql@provide@backup@amssub{alignedat}%
5468   \eql@provide@backup@amssub{gathered}%
5469   \eql@provide@backup@multlined
5470   \eql@provide@backup@subequations
5471 }

```

Replacement amsmath Environments. **TODO:** describe

```

5472 \def\eql@alignat@gobblecol#1{%
5473   \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.)
5474 \newenvironment{eql@gathered}
5475   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5476 \newenvironment{eql@multlined}
5477   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5478 \newenvironment{eql@aligned}
5479   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
5480 \newenvironment{eql@alignedat}
5481   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
5482   {\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.*)

```

5483 \newenvironment{eql@equation}
5484   {\eqnaddopt{equation}\equations}\endequations}
5485 \newenvironment{eql@displaymath}
5486   {\eqnaddopt{equation,nonumber}\equations}\endequations}
5487 \newenvironment{eql@gather}
5488   {\eqnaddopt{lines}\equations}\endequations}
5489 \newenvironment{eql@multline}
5490   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5491   {\endequations}
5492 \newenvironment{eql@align}
5493   {\eqnaddopt{columns}\equations}\endequations}
5494 \newenvironment{eql@flalign}
5495   {\eqnaddopt{fulllength}\eql@align}\endequations}
5496 \newenvironment{eql@alignat}
5497   {\eqnaddopt{colsep=off}\eql@xalignat}\endequations}
5498 \newenvironment{eql@xalignat}
5499   {\eql@alignat@gobblecol\eql@align}\endequations}
5500 \newenvironment{eql@xxalignat}
5501   {\eqnaddopt{fulllength}\eql@xalignat}\endequations}
5502 \newenvironment{eql@equation*}
5503   {\eqnaddopt{nonumber}\eql@equation}\endequations}
5504 \newenvironment{eql@gather*}
5505   {\eqnaddopt{nonumber}\eql@gather}\endequations}
5506 \newenvironment{eql@multline*}
5507   {\eqnaddopt{nonumber}\eql@multline}\endequations}
5508 \newenvironment{eql@align*}
5509   {\eqnaddopt{nonumber}\eql@align}\endequations}
5510 \newenvironment{eql@flalign*}
5511   {\eqnaddopt{nonumber}\eql@flalign}\endequations}
5512 \newenvironment{eql@alignat*}
5513   {\eqnaddopt{nonumber}\eql@alignat}\endequations}
5514 \newenvironment{eql@xalignat*}
5515   {\eqnaddopt{nonumber}\eql@xalignat}\endequations}

```

Install Additional Environments. The additional environments need to be installed at their intended names which can be adjusted by the user.

`eql@provide@onlyonce` Process arguments for providing a specific environment. #1 describes the environment using the `amsmath` name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the `amsmath` environment in place making sure that the replacement is robust against loading `amsmath` before or after. If #2 equals ‘*’, just overwrite the `amsmath` environment in place immediately (e.g. within a block in the document body):

```

5516 \def\eql@provide@onlyonce#1#2{%
5517   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
5518   \ifx\eql@tmp\eql@tmpa
5519     \let\eql@tmp\@empty
5520   \fi
5521   \ifx\eql@tmp\@empty
5522     \let\eql@tmp\@undefined
5523     \ifx\@nodocument\relax
5524       \def\eql@tmp{#1}%
5525     \fi
5526     \ifcsname eql@provided@#1\endcsname

```

```

5527     \def\eql@tmp{#1}%
5528     \fi
5529     \eql@letcs{eql@provided@#1}\eql@true
5530 \else
5531     \def\eql@tmpa{*}%
5532     \ifx\eql@tmp\eql@tmpa
5533         \def\eql@tmp{#1}%
5534     \fi
5535 \fi
5536 }

```

\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:

```

5537 \def\eql@provide@eqref#1{%
5538     \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
5539     \ifx\eql@tmp\eql@tmpa
5540         \let\eql@tmp\@empty
5541     \fi
5542     \ifx\eql@tmp\@empty
5543         \eql@provide@onlyonce{eqref}{}%
5544     \else
5545         \def\eql@tmpa{*}%
5546         \ifx\eql@tmp\eql@tmpa
5547             \def\eql@tmp{eqref}%
5548         \else
5549             \edef\eql@tmp{\expandafter\@gobble\string#1}%
5550         \fi
5551     \fi
5552     \ifdefined\eql@tmp
5553         \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
5554     \else
5555         \eql@amsmath@after{%
5556             \eql@provide@movecmd{eqref}{eql@eqref}%
5557         }%
5558         \AddToHook{package/mathtools/after}{%
5559             \eql@provide@movecmd{eqref}{eql@eqref}%
5560         }%
5561         \eql@provide@movecmd{eqref}{eql@eqref}%
5562         \eql@amsmath@undefine\eqref
5563     \fi
5564 }

```

\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:

```

5565 \def\eql@provide@amsenv#1#2{%
5566     \eql@provide@onlyonce{#1}{#2}%
5567     \ifdefined\eql@tmp
5568         \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5569         \eql@tagging@register@luamml{\eql@tmp}%
5570         \eql@markline@amsthm@register{\eql@tmp}%
5571     \else
5572         \eql@amsmath@after{%
5573             \eql@provide@moveenv{#1}{eql@#1}%
5574             \eql@markline@amsthm@register{#1}%
5575         }%

```

```

5576 \AddToHook{package/mathtools/after}{%
5577 \eql@provide@moveenv{#1}{eql@#1}%
5578 \eql@markline@amsthm@register{#1}%
5579 }%
5580 \eql@provide@moveenv{#1}{eql@#1}%
5581 \eql@markline@amsthm@register{#1}%
5582 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
5583 \fi
5584 }

```

\eql@provide@amssub 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:

```

5585 \def\eql@provide@amssub#1#2{%
5586 \eql@provide@onlyonce{#1}{#2}%
5587 \ifdefined\eql@tmp
5588 \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5589 \else
5590 \eql@amsmath@after{%
5591 \eql@provide@moveenv{#1}{eql@#1}}%
5592 \AddToHook{package/mathtools/after}{%
5593 \eql@provide@moveenv{#1}{eql@#1}}%
5594 \eql@provide@moveenv{#1}{eql@#1}%
5595 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
5596 \fi
5597 }

```

\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:

```

5598 \def\eql@provide@multlined#1{%
5599 \eql@provide@onlyonce{multlined}{#1}%
5600 \ifdefined\eql@tmp
5601 \eql@provide@moveenv{\eql@tmp}{eql@multlined}%
5602 \else
5603 \AddToHook{package/mathtools/after}{%
5604 \eql@provide@moveenv{multlined}{eql@multlined}}%
5605 \eql@provide@moveenv{multlined}{eql@multlined}%
5606 \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5607 \eql@provide@undefineenv{multlined}}}%
5608 \fi
5609 }

```

\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:

```

5610 \def\eql@provide@equation#1{%
5611 \eql@provide@onlyonce{equation}{#1}%
5612 \ifdefined\eql@tmp
5613 \eql@provide@moveenv{\eql@tmp}{eql@equation}%
5614 \eql@tagging@register@luamml{\eql@tmp}%
5615 \eql@markline@amsthm@register{\eql@tmp}%
5616 \else
5617 \eql@amsmath@after{%
5618 \eql@provide@moveenv{equation}{eql@equation}%

```



```

5619     \eql@markline@amsthm@register{equation}%
5620 }%
5621 \AddToHook{package/hyperref/after}{%
5622     \ifpackageloaded{amsmath}{}%
5623     \eql@provide@moveenv{equation}{eql@equation}%
5624     \eql@markline@amsthm@register{equation}%
5625 }%
5626 }%
5627 \eql@provide@moveenv{equation}{eql@equation}%
5628 \eql@markline@amsthm@register{equation}%
5629 \ifdefined\eql@tagging@on
5630     \AddToHook{begindocument/end}{%
5631         \eql@provide@moveenv{equation}{eql@equation}%
5632         \eql@markline@amsthm@register{equation}%
5633     }%
5634 \fi
5635 \fi
5636 }

```

`\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:

```

5637 \def\eql@provide@equationstar#1{%
5638     \eql@provide@onlyonce{equation*}{#1}%
5639     \ifdefined\eql@tmp
5640         \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
5641         \eql@tagging@register@luamml{\eql@tmp}%
5642         \eql@markline@amsthm@register{\eql@tmp}%
5643     \else
5644         \eql@amsmath@after{%
5645             \eql@provide@moveenv{equation*}{eql@equation*}%
5646             \eql@markline@amsthm@register{equation*}%
5647         }%
5648         \eql@provide@moveenv{equation*}{eql@equation*}%
5649         \eql@markline@amsthm@register{equation*}%
5650         \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5651         \ifdefined\eql@tagging@on
5652             \AddToHook{begindocument/end}{%
5653                 \eql@provide@moveenv{equation*}{eql@equation*}%
5654                 \eql@markline@amsthm@register{equation*}%
5655             }%
5656         \fi
5657     \fi
5658 }

```

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

```

5659 \def\eql@provide@displaymath#1{%
5660     \eql@provide@onlyonce{displaymath}{#1}%
5661     \ifdefined\eql@tmp
5662         \eql@provide@moveenv{\eql@tmp}{eql@displaymath}%
5663         \eql@markline@amsthm@register{\eql@tmp}%
5664         \eql@tagging@register@luamml{\eql@tmp}%
5665     \else
5666         \eql@provide@moveenv{displaymath}{eql@displaymath}%
5667         \eql@markline@amsthm@register{displaymath}%
5668     \ifdefined\eql@tagging@on

```

```

5669      \AddToHook{begindocument/end}{%
5670        \eql@provide@moveenv{displaymath}{eql@displaymath}}}%
5671    \fi
5672  \fi
5673 }

```

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:

```

5674 \def\eql@provide@subequations#1{%
5675   \eql@provide@onlyonce{subequations}{#1}%
5676   \ifdefined\eql@tmp
5677     \eql@provide@moveenv{\eql@tmp}{eql@subequations}%
5678   \else
5679     \eql@amsmath@after{%
5680       \eql@provide@moveenv{subequations}{eql@subequations}%
5681     }%
5682     \AddToHook{package/hyperref/after}{%
5683       \AddToHook{cmd/subequations/before}[hyperref]{}%
5684       \AddToHook{cmd/subequations/after}[hyperref]{}%
5685       \RemoveFromHook{cmd/subequations/before}[hyperref]%
5686       \RemoveFromHook{cmd/subequations/after}[hyperref]%
5687       \AddToHook{begindocument/end}{%
5688         \eql@provide@moveenv{subequations}{eql@subequations}}}%
5689     }%
5690     \eql@provide@moveenv{subequations}{eql@subequations}%
5691     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}}%
5692   \fi
5693 }

```

\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:

```

5694 \def\eql@provide@sqr{%
5695   \let\[\eql@sqr@open
5696   \let\]\eql@sqr@close
5697   \eql@amsmath@after{%
5698     \let\[\eql@sqr@open
5699     \let\]\eql@sqr@close
5700   }%
5701   \ifdefined\eql@tagging@on
5702     \AddToHook{begindocument/end}{%
5703       \let\[\eql@sqr@open
5704       \let\]\eql@sqr@close
5705     }%
5706   \fi
5707 }

```

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

```

5708 \def\eql@provide@ang{%
5709   \let\<\eql@ang@open

```

```

5710 \let\>\eql@ang@close
5711 }

```

Interface.

provide (*key*) We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

5712 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5713 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
5714 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5715 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}
5716 \eql@define@key{provide}{multline}[]{\eql@provide@amsenv{multline}{#1}}
5717 \eql@define@key{provide}{align}[]{\eql@provide@amsenv{align}{#1}}
5718 \eql@define@key{provide}{flalign}[]{\eql@provide@amsenv{flalign}{#1}}
5719 \eql@define@key{provide}{alignat}[]{\eql@provide@amsenv{alignat}{#1}}
5720 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsenv{xalignat}{#1}}
5721 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsenv{xxalignat}{#1}}
5722 \eql@define@key{provide}{aligned}[]{\eql@provide@amssub{aligned}{#1}}
5723 \eql@define@key{provide}{alignedat}[]{\eql@provide@amssub{alignedat}{#1}}
5724 \eql@define@key{provide}{gather*}[]{\eql@provide@amsenv{gather*}{#1}}
5725 \eql@define@key{provide}{multline*}[]{\eql@provide@amsenv{multline*}{#1}}
5726 \eql@define@key{provide}{align*}[]{\eql@provide@amsenv{align*}{#1}}
5727 \eql@define@key{provide}{flalign*}[]{\eql@provide@amsenv{flalign*}{#1}}
5728 \eql@define@key{provide}{alignat*}[]{\eql@provide@amsenv{alignat*}{#1}}
5729 \eql@define@key{provide}{xalignat*}[]{\eql@provide@amsenv{xalignat*}{#1}}
5730 \eql@define@key{provide}{gathered}[]{\eql@provide@amssub{gathered}{#1}}
5731 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5732 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5733 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
5734 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
5735 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
5736 \eql@define@key{provide}{tagform}[]{%
5737   \def\tagform@##1{\maketag@@@{\eql@tags@tagform{#1}}}}
5738 \eql@define@key{provide}{maketag}[]{%
5739   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}

```

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

```

5740 \newcommand{\eqnlinesprovide}[1]{%
5741   \eql@setkeys{provide}{#1}%
5742   \ignorespaces
5743 }

```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```

5744 \let\eql@error@packageoption@gobble

```

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

```

5745 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
5746 \DeclareOption{fleqn}{\eqnlineset{left}}

```

Pass undeclared options on to keyval processing:

```

5747 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}

```

Set defaults for package:

```
5748 \eqldefaults@eqnlines
5749 \eql@mode@columns
5750 \eql@mode@aligned
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the \TeX conditional parsing mechanism:

```
5751 \ifdefined\tagsleft@true\else
5752   \expandafter\newif\csname iftagsleft@\endcsname
5753 \fi
5754 \ifdefined\@fleqntrue\else
5755   \expandafter\newif\csname if@fleqn\endcsname
5756 \fi
```

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

```
5757 \eql@amsmath@after{%
5758   \ifnum\eql@provide@opt@env=\tw@
5759     \iftagsleft@
5760       \eqnlineset{tags=left}%
5761     \else
5762       \eqnlineset{tags=right}%
5763     \fi
5764     \if@fleqn
5765       \eqnlineset{layout=left}%
5766     \else
5767       \eqnlineset{layout=center}%
5768     \fi
5769   \fi
5770 }
```

Process package options:

```
5771 \ProcessOptions
```

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

```
5772 \def\eql@error@packageoption#1{%
5773   \eql@error{may only use '#1' as a package option}%
5774 }
```

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

```
5775 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

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

```
5776 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

Provide native \LaTeX environment `equation` and symbolic shortcut `\[...\]` if desired:

```
5777 \ifnum\eql@provide@opt@env>\z@
5778   \eqnlinesprovide{equation,equation*,sqr,displaymath}
5779 \fi
```

Provide `amsmath` equation environments if desired:

```
5780 \ifnum\eql@provide@opt@env=\tw@
5781   \eqnlinesprovide{%
```

```

5782    multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
5783    multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
5784    multlined,gathered,aligned,alignedat,%
5785    subequations}
5786 \fi

```

Provide symbolic shortcut `\<...\>` if desired:

```

5787 \ifdefined\eqL@provide@opt@ang\eqnlinesprovide{ang}\fi

```

Provide equation reference `\eqref` if desired:

```

5788 \ifdefined\eqL@provide@opt@eqref\eqnlinesprovide{eqref}\fi

```