

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`

2025/12/28, v0.12.1

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

Abstract

eqnlines is a L^AT_EX 2_ε package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of L^AT_EX 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	Development 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	amsmath Integration	8
2.7	PDF Tagging Support	8
2.8	Key-Value Processing	9
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	20
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	31
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	50
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	70
9.5	Tag Margin	72
9.6	Single Column	72
9.7	Multiple Columns	73
10	Single Column Arrangement	79
10.1	Supporting Definitions	79
10.2	Arrangement Methods	79
10.3	Central Alignment	81
10.4	Left Alignment	82
10.5	Right Alignment	83
11	Equations Box Environment	85
11.1	Line Breaks	85
11.2	Stacked Mode	85
11.3	Aligned Mode	86
11.4	Main	88
11.5	Environment	90
12	Single-Line Equation	90

12.1 Native Mode	90
12.2 Print	91
13 Multi-Line with Single Column	92
13.1 Measure	92
13.2 Column Placement	94
13.3 Print	94
14 Multi-Line with Multiple Columns	96
14.1 Support	96
14.2 Transpose	96
14.3 Measure	98
14.4 Columns Placement	101
14.5 Print	101
15 Interface	105
15.1 Scanning the Equation Body	105
15.2 Options Processing	108
15.3 Single-Line Main	109
15.4 Multi-Line Main	111
15.5 Equations Environment	112
16 Options	115
16.1 Selection Tools	115
16.2 Options Declarations	116
16.3 Parameter Presets	127
16.4 Component Selection	129
16.5 Global and Package Options	137

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 Development Messages

The package offers a version `eqnlines-dev` for development and debugging purposes. It outputs extra information on the current location within the code in order to track

progress. The extra lines for the development version are indicated as ‘(dev)’ in the implementation documentation:

```

1 (dev)\def\eql@dev#1{\PackageInfo{eqnlines-dev}{#1}}
2 (dev)\def\eql@dev@start#1{\eql@dev{starting \string#1}}
3 (dev)\def\eql@dev@enter#1{\eql@dev{entering \string#1}}
4 (dev)\def\eql@dev@leave#1{\eql@dev{ leaving \string#1}}
5 (dev)\def\eql@dev@enterenv{\eql@dev{entering \@currenvir}}
6 (dev)\def\eql@dev@leaveenv{\eql@dev{ leaving \@currenvir}}
7 (dev)\def\eql@dev@in#1#2{\eql@dev{ \space within \string#1 #2}}

```

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:

```

8 \let\eql@false\@undefined
9 \let\eql@true\@empty

```

TODO: describe

```

10 \def\eql@append#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
11 \def\eql@appendexpand#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
12 \def\eql@appendmacro#1#2{\eql@appendexpand#1{\unexpanded\expandafter{#2}}}
13 \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:

```

14 \ifdefined\dollar@begin
15 \def\eql@dollar@begin{\dollar@begin}
16 \def\eql@dollar@end{\dollar@end}
17 \else
18 \def\eql@dollar@begin{\$}
19 \def\eql@dollar@end{\$}
20 \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.

`\eql@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 `\eql@ifnextchar@loose`. We replicate the `amsgen` version `\new@ifnextchar` that does not skip over spaces as `\eql@ifnextchar@loose`. The space before #1 allows to look-ahead for spaces as well:

```

21 \let\eq@ifnextchar@loose\kernel@ifnextchar
22 \long\def\eq@ifnextchar@tight#1#2#3{%
23   \let\reserved@d= #1%
24   \def\reserved@a{#2}%
25   \def\reserved@b{#3}%
26   \futurelet\@let@token\eq@ifnch@tight
27 }
28 \def\eq@ifnch@tight{%
29   \ifx\@let@token\reserved@d
30     \let\reserved@b\reserved@a
31   \fi
32   \reserved@b
33 }

```

`\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:

```

34 \let\eq@atxi=@
35 \begingroup
36   \makeatother
37   \let\tmp=@%
38   \makeatletter
39   \global\let\eq@atxii\tmp
40 \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`, `\eq@teststaropt@...` 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 [...]:

```

41 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
42 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
43 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
44 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
45 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{%
46   \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
47 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{%
48   \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}}
49 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[#{1}]{#1[#{2}]}%]
50 \long\def\eq@testopt@tight#1#2{\eq@ifnextchar@tight[#{1}]{#1[#{2}]}%]
51 \long\def\eq@teststaropt@loose#1#2#3{%
52   \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}}
53 \long\def\eq@teststaropt@tight#1#2#3{%
54   \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}}
55 \long\def\eq@teststaroropt@loose#1#2#3{%
56   \eq@ifstar@loose{#1}{\eq@testopt@loose{#2}{#3}}}
57 \long\def\eq@teststaroropt@tight#1#2#3{%
58   \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}}
59 \long\def\eq@gobbleopt[#1]{#1}
60 \long\def\eq@gobbleoptone[#1]#2{#1}

```

TODO: describe

```

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

```

TODO: describe

```

62 \def\eql@parseopt#1#2{%
63   \def\eql@parseopt@case{#1}%
64   \def\eql@parseopt@end{#2}%
65   \eql@parseopt@peek
66 }
67 \def\eql@parseopt@peek{%
68   \futurelet\eql@parseopt@token\eql@parseopt@select
69 }
70 \def\eql@parseopt@select{%
71   \let\eql@parseopt@next\eql@parseopt@other
72   \ifx\eql@parseopt@token\@sptoken
73     \let\eql@parseopt@next\eql@parseopt@end
74   \fi
75   \eql@parseopt@case
76   \eql@parseopt@next
77 }
78 \def\eql@parseopt@other{\eql@parseopt@warn\eql@parseopt@end}
79 \let\eql@parseopt@warn\@empty
80 \def\eql@parseopt@gobble#1{\eql@parseopt@peek}

```

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

```

81 \def\eql@spbgroup{\iffalse{\fi\ifnum0='}\fi}
82 \def\eql@speggroup{\ifnum0='{ \fi\iffalse}\fi}
83 \def\eql@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
84 \def\eql@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}

```

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

```

85 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1\eql@sregroup#2}
86 \long\def\eql@ampprotecttwo#1#2#3{%
87   \eql@srbgroup#1\eql@sregroup#2\eql@sregroup#3}

```

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

```

88 \def\eql@ifnextchar@loose@ampsafe#1{%
89   \eql@ampprotecttwo\eql@ifnextchar@loose#1}
90 \def\eql@ifnextchar@tight@ampsafe#1{%
91   \eql@ampprotecttwo\eql@ifnextchar@tight#1}
92 \def\eql@ifstar@loose@ampsafe\eql@ampprotecttwo\eql@ifstar@loose}
93 \def\eql@ifstar@tight@ampsafe\eql@ampprotecttwo\eql@ifstar@tight}
94 \def\eql@testopt@loose@ampsafe\eql@ampprotect\eql@testopt@loose}
95 \def\eql@testopt@tight@ampsafe\eql@ampprotect\eql@testopt@tight}
96 \def\eql@teststaropt@loose@ampsafe\eql@ampprotecttwo\eql@teststaropt@loose}
97 \long\def\eql@teststaropt@tight@ampsafe{%
98   \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

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

optional arguments and starred variants more robust in this context. The macro `\eql@amproof` overwrites the original definitions, and `\eql@amprevert` reverts the changes:

```

99 \let\eql@ifnextchar@org@ifnextchar
100 \let\eql@ifstar@org@ifstar
101 \let\eql@testopt@org@testopt
102 \def\eql@amprevert{%
103   \let@ifnextchar\eql@ifnextchar@org
104   \let@testopt\eql@testopt@org
105   \let@ifstar\eql@ifstar@org
106 }
107 \def\eql@ampproof{%
108   \let@ifnextchar\eql@ifnextchar@loose@ampsafe
109   \let@testopt\eql@testopt@loose@ampsafe
110   \let@ifstar\eql@ifstar@loose@ampsafe
111 }

```

2.5 Error Messages

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

```

112 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
113 \def\eql@warning{\PackageWarning{eqnlines}}

```

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

```

114 \def\eql@warn@here#1{\eql@warning{\string#1 not allowed outside equations}}
115 \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:

```

\warn@label@multiple
\warn@label@multiple
\warn@tag@unused
\warn@tag@multiple
\warn@name@unused
\warn@name@multiple
\warn@ref@unused
\warn@ref@multiple
116 \def\eql@warn@tags@unused#1#2{\eql@warning{Unused equation #1:
117   #2 will be lost}}
118 \def\eql@warn@tags@multiple#1#2#3{\eql@warning{Multiple equation #1:
119   previous #2 will be lost#3}}
120 \def\eql@warn@label@unused{\eql@warn@tags@unused{\string\label}
121   {label '\eql@tags@label'}}
122 \def\eql@warn@label@multiple#1{\eql@warn@tags@multiple{\string\label's}
123   {label '\eql@tags@label'}}{ and replaced by '#1'}}
124 \def\eql@warn@name@unused{\eql@warn@tags@unused{label name}
125   {name declaration}}
126 \def\eql@warn@name@multiple{\eql@warn@tags@multiple{label names}
127   {name declaration}{}}
128 \def\eql@warn@tag@unused{\eql@warn@tags@unused{\string>tag}
129   {tag declaration}}
130 \def\eql@warn@tag@multiple{\eql@warn@tags@multiple{\string>tag's}
131   {tag declaration will be lost}{}}
132 \def\eql@warn@ref@unused{\eql@warn@tags@unused{tag label}
133   {tag label declaration}}
134 \def\eql@warn@ref@multiple{\eql@warn@tags@multiple{tag labels}
135   {tag label declaration}{}}

136 \def\eql@warn@parseopt{%
137   \eql@warning{Unknown modifier token: starting math content}}
138 \def\eql@warn@parseopt@verbose{%
139   \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`
`\eql@amsmath@before` executes argument #1 after loading `amsmath` is loaded. It also runs the code if `amsmath`
`\eql@amsmath@undefine` has already been loaded. Furthermore, loading `amsmath` requires certain macros to be
`\eql@amsmath@let` 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/`:

```
140 \def\eql@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
141 \def\eql@amsmath@before#1{%
142   \ifpackageloaded{amsmath}{\AddToHook{package/amsmath/before}{#1}}
143 \def\eql@amsmath@undefine#1{\eql@amsmath@before{\let#1\undefined}}
144 \def\eql@amsmath@let#1#2{\eql@amsmath@undefine#1\let#1#2}
```

TODO: temporary fix for development stages

```
145 \@ifpackageloaded{amsmath}{}{
146   \DeclareHookRule{package/amsmath/after}
147     {eqlines}{after}{latex-lab-testphase-math}}
```

2.7 PDF Tagging Support

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

```
148 \let\eql@tagging@on\eql@false
149 \IfFormatAtLeastTF{2025-06-01}{%
150   \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}{}
151 \ifdefined\eql@tagging@on
152   \def\eql@tagging@mathsave{%
153     \UseTaggingSocket{math/luamml/save/nNn}{{}\displaystyle{mtd}}}
154   \def\eql@tagging@mathaddlast{%
155     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
156   \def\eql@tagging@tagbegin{%
157     \UseTaggingSocket{math/display/tag/begin}}
158   \def\eql@tagging@tagend{%
159     \UseTaggingSocket{math/display/tag/end}}
160   \def\eql@tagging@tagsave{%
161     \UseTaggingSocket{math/luamml/mtable/tag/save}}
162   \def\eql@tagging@tagaddbox{%
163     \setbox\z@\copy\eql@tagbox%
164     \UseTaggingSocket{math/luamml/mtable/tag/set}}
165   \def\eql@tagging@tablesaveinner{%
166     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
167   \def\eql@tagging@tableaddinner{%
168     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
169   \def\eql@tagging@tablesavelines{%
170     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
171   \def\eql@tagging@tablesavealign{%
172     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
173   \def\eql@tagging@alignleft{%
174     \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
175   \def\eql@tagging@aligncenter{%
176     \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
```

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


```

177 \def\eql@tagging@alignright{%
178   \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:

```

179 \let\eql@single@doscan\eql@true
180 \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:

```

```

181 \def\eql@tagging@start{%
182   \csname bool_if:N\expandafter\endcsname
183   \csname l__math_collected_bool\endcsname{%
184     \edef\eql@tmp{\@currenvir}{\unexpanded\expandafter{\eql@tagging@opt}}%
185     \the\eql@scan@reg@}%
186     \csname __math_process:nn\expandafter\endcsname\eql@tmp
187     \@kernel@math@registered@begin
188     \csname bool_set_true:N\expandafter\endcsname
189     \csname l__math_collected_bool\endcsname
190   }%
191 }
192 \def\eql@tagging@end{}
193 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}
194 \else
195 \def\eql@tagging@mathsave{}
196 \def\eql@tagging@mathaddlast{}
197 \def\eql@tagging@tagbegin{}
198 \def\eql@tagging@tagend{}
199 \def\eql@tagging@tagsave{}
200 \def\eql@tagging@tagaddbox{}
201 \def\eql@tagging@tablesaveinner{}
202 \def\eql@tagging@tableaddinner{}
203 \def\eql@tagging@tablesavealign{}
204 \def\eql@tagging@tablesavealign{}
205 \def\eql@tagging@alignleft{}
206 \def\eql@tagging@aligncenter{}
207 \def\eql@tagging@alignright{}
208 \def\eql@tagging@start{}
209 \def\eql@tagging@end{}
210 \def\eql@tagging@register@env{\@gobble}
211 \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:

```

212 \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:

```

213 \def\eql@decide@relax{\eql@tmpb:=\relax}
214 \def\eql@decide@select#1#2#3#4{%
215   \def\eql@tmpa{#3}%
216   \let\eql@tmpd\undefined
217   \@for\eql@tmpc:=#4\do{%
218     \ifdefined\eql@tmpd\else
219       \edef\eql@tmpb{\noexpand\eql@tmpb:=\expandafter\@firstoftwo\eql@tmpc}%
220       \ifx\eql@tmpb\eql@decide@relax
221         \def\eql@tmpa{\relax}%
222       \fi
223       \expandafter\@for\eql@tmpb\do{%
224         \ifx\eql@tmpa\eql@tmpb
225           \edef\eql@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
226             \expandafter\@secondoftwo\eql@tmpc}}%
227         \fi
228       }%
229     \fi
230   }%
231   \ifdefined\eql@tmpd
232     \eql@tmpd
233   \else
234     \eql@error{undefined value '#3' for option '#2' of '#1'}%
235   \fi
236 }

```

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

```

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

```

`\eql@decide@if`

```

239 \def\eql@decide@if#1#2#3#4#5{%
240   \eql@decide@select{#1}{#2}{#3}{%
241     {\eql@decide@true{#4}},%
242     {\eql@decide@false{#5}}}%

```

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

```

243 \def\eql@decide@bool#1#2#3#4{%
244   \eql@decide@if{#1}{#2}{#3}{\let#4\eql@true}{\let#4\eql@false}}

```

Key Declaration.

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

```

245 \def\eql@define@key#1#2{%
246   \eql@ifnextchar@loose[%]
247     {\eql@definekey@opt{#1}{#2}}%
248     {\eql@definekey@noopt{#1}{#2}}%
249 }
250 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{#3}}
251 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{#3}{#4}}
252 \def\eql@definekey@for#1#2#3{%
253   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}{#3}}%
254   \edef\eql@for@vara{\noexpand\eql@for@vara:=#1}%
255   \expandafter\@for\eql@for@vara\do{%
256     \edef\eql@for@varb{\noexpand\eql@for@varb:=#2}%
257     \expandafter\@for\eql@for@varb\do{%
258       \edef\eql@for@call##1{%
259         \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@vara}}%
260       \eql@for@call{##1}}%
261     }%
262   }%
263 }

```

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

```

264 \def\eql@setkeys#1#2{%
265   \def\eql@tmp{\setkeys{eql@#1}}%
266   \expandafter\eql@tmp\expandafter{#2}%
267 }

```

Options and Control Interface.

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

```

268 \let\eql@nextopt\@empty
269 \def\eql@nextopt@process#1{%
270 (dev)\eql@dev@start\eql@nextopt@process
271   \eql@setkeys{#1}\eql@nextopt
272   \let\eql@tagging@opt\eql@nextopt
273   \global\let\eql@nextopt\@empty
274 }

```

`\eqnaddopt`

```

275 \newcommand{\eqnaddopt}[1]{%
276   \ifx\eql@nextopt\@empty
277     \eql@append\eql@nextopt{#1}%
278   \else
279     \eql@append\eql@nextopt{, #1}%
280   \fi
281 }

```

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

```

282 \newcommand{\eqnlineset}[1]{%
283 (dev)\eql@dev@start\eqnlineset
284   \eql@setkeys{setup}{#1}%
285   \ignorespaces
286 }

```

`\eql@control@default`

```
287 \protected\def\eql@control@default{%
288   \eql@warn@here\eqncontrol
289   \@gobble
290 }
291 \let\eqncontrol\eql@control@default
```

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

```
292 \newcommand{\eql@control}[1]{%
293   \relax
294   \eql@setkeys{control}{#1}%
295   \ignorespaces
296 }
```

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:

```
297 \let\eql@tagsleft\eql@false
```

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

```
298 \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`:

```
299 \def\eql@layoutleftmargin{\leftmargini}
300 \def\eql@layoutleftmarginmax{.5\maxdimen}
301 \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

```
302 \newdimen\eql@tagmargin@
303 \let\eql@tagmargin@val\undefined
304 \newdimen\eql@tagmargin@ratio@
305 \eql@tagmargin@ratio@\p@
306 \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:

```
307 \newdimen\eql@indent@
308 \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:

```
309 \newdimen\eql@paddingleft@
310 \newdimen\eql@paddingright@
311 \let\eql@paddingleft@val\undefined
312 \let\eql@paddingright@val\undefined
```

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

```
\eql@display@marginleft
\eql@display@marginright
313 \let\eql@display@linewidth\undefined
314 \let\eql@display@marginleft\undefined
315 \let\eql@display@marginright\undefined
```

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

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

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

```
\eql@break@line@shortsep
\eql@break@col@sep
\eql@break@col@shortsep
317 \def\eql@break@line@sep{2em minus 1em}
318 \def\eql@break@line@shortsep{1em}
319 \def\eql@break@col@sep{2em minus 1em}
320 \def\eql@break@col@shortsep{1em}
```

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

```
321 \def\eql@spread@val{\jot}
322 \newdimen\eql@spread@
```

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

```
323 \newdimen\eql@tagfuzz@
324 \eql@tagfuzz@16sp\relax
```

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

```
325 \def\eql@display@height\undefined
326 \def\eql@display@depth\undefined
```

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

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

```
328 \def\eq@skip@mode@cont@above{2}
329 \def\eq@skip@mode@cont@below{0}
```

```
330 \def\eq@skip@mode@par@above{3}
331 \def\eq@skip@mode@par@below{0}
```

```
332 \def\eq@skip@mode@top@above{4}
333 \def\eq@skip@mode@top@below{0}
```

```
334 \newcount\eq@skip@mode@leave@
335 \let\eq@skip@force@leave@\undefined
```

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

```
\eq@skip@force@below
\mode@above@ (counter) 336 \newcount\eq@skip@mode@above@
\mode@below@ (counter) 337 \newcount\eq@skip@mode@below@
338 \let\eq@skip@force@above@\undefined
339 \let\eq@skip@force@below@\undefined
340 \let\eq@skip@custom@above@\undefined
341 \let\eq@skip@custom@below@\undefined
```

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

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

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

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

```
342 \def\eq@skip@long@above{\abovedisplayskip}
343 \def\eq@skip@long@below{\belowdisplayskip}
344 \def\eq@skip@short@above{\abovedisplayshortskip}
345 \def\eq@skip@short@below{\belowdisplayshortskip}
346 \def\eq@skip@cont@above{\eq@skip@short@above}
347 \def\eq@skip@cont@below{\eq@skip@short@below}
348 \def\eq@skip@par@above{\eq@skip@long@above}
349 \def\eq@skip@par@below{\eq@skip@long@below}
350 \def\eq@skip@top@above{\eq@skip@long@above}
351 \def\eq@skip@top@below{\eq@skip@long@below}
352 \def\eq@skip@med@above{\abovedisplayskip/2}
353 \def\eq@skip@med@below{\belowdisplayskip/2}
354 \def\eq@skip@tag@above{\z@skip}
355 \def\eq@skip@tag@below{\z@skip}
```

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

```
356 \newdimen\eq@colsepmin@
357 \def\eq@colsepmin@val{1em}
358 \def\eq@colsepmax@val{.5\maxdimen}
```

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

```

359 \newdimen\eq\tagwidthmin@
360 \eq\tagwidthmin@\z@

```

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

```

361 \newdimen\eq\tagsepmin@
362 \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
363 \def\eq\equations@sqr@opt{equation,nonumber}
364 \def\eq\equations@ang@opt{align,nonumber}
365 \def\eq\box@ang@opt{align}

```

Multi-Line Align Mode.

```

366 \let\eq\columns@fulllength\eq>false

```

3.2 Registers

TODO: describe

General. **TODO:** describe

```

367 \newcount\eq\count@
368 \newdimen\eq\dimen@
369 \newskip\eq\skip@

```

TODO: describe

```

370 \let\eq\display@container\@empty

```

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

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

`\eq\cellwidth@` (*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*)

```

371 \newbox\eq\cellbox@
372 \newbox\eq\tagbox@
373 \newdimen\eq\cellwidth@
374 \newdimen\eq\prevwidth@
375 \newdimen\eq\tagwidth@
376 \newdimen\eq\prevdepth@
377 \newcount\eq\prevgraf@

```

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

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

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

```

378 \newdimen\eq\totalwidth@
379 \newdimen\eq\tagwidth@max@
380 \newdimen\eq\totalheight@
381 \newdimen\eq\topheight@
382 \newdimen\eq\bottomdepth@

```

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

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

```
383 \newdimen\eq@line@height@
384 \newdimen\eq@line@depth@
```

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

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

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

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

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

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

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

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

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

```
385 \newdimen\eq@line@width@
386 \newdimen\eq@line@avail@
387 \newdimen\eq@line@pos@
388 \newcount\eq@line@availsep@
389 \newcount\eq@line@widthsep@
390 \newcount\eq@line@possep@
391 \newdimen\eq@line@offset@
392 \newdimen\eq@line@prevdepth@
393 \newdimen\eq@line@interline@
```

Rows and Columns.

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

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

```
394 \newcount\eq@row@
395 \newcount\eq@totalrows@
396 \newcount\eq@tagrows@
```

`\eq@column@`

`\eq@totalcolumns@`

```
397 \newcount\eq@column@
398 \newcount\eq@totalcolumns@
```

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

```
399 \newdimen\eq@colsep@
```

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

```
400 \newcount\eq@intercolumns@
```

Vertical Spacing Adjustments.

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

```
401 \newdimen\eq@display@firstavail@
402 \def\eq@display@firstavail@set#1{%
403   \ifnum\eq@row@=\@ne
404     \global\eq@appendexpand\eq@display@container{%
405       \eq@display@firstavail@\the#1\relax}%
406   \fi
407 }
```


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

```
408 \newdimen\eqldisplay@aboveextend@
409 \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:

```
410 \ifdefined\measuring@true\else
411   \expandafter\newif\csname ifmeasuring@\endcsname
412 \fi
413 \AddToHook{package/hyperref/after}{
414   \ifdefined\Hy@ifnotmeasuring
415     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
416   \fi
417 }
```

`\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:

```
418 \ifdefined\@displaytrue\else
419   \expandafter\newif\csname if@display\endcsname
420   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
421 \fi
```

3.3 Hooks

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

```
422 \let\eql@hook@blockbefore\@empty
423 \let\eql@hook@blockafter\@empty
424 \let\eql@hook@blockin\@empty
425 \let\eql@hook@blockout\@empty
426 \let\eql@hook@linein\@empty
427 \let\eql@hook@lineout\@empty
428 \let\eql@hook@colin\@empty
429 \let\eql@hook@colout\@empty
430 \let\eql@hook@eqin\@empty
431 \let\eql@hook@eqout\@empty
432 \let\eql@hook@innerleft\@empty
433 \let\eql@hook@innerright\@empty
434 \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

```
435 \let\eql@punct@col\@empty
436 \let\eql@punct@line\relax
437 \let\eql@punct@block\relax
438 \let\eql@punct@main\relax
```

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

```
439 \let\eql@punct@sep\relax
```

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

```
440 \def\eql@punct@relax{\relax}
441 \def\eql@punct@tilde{~}
442 \def\eql@punct@set#1#2{%
443   \def#1{#2}%
444   \ifx#1\eql@punct@relax
445     \let#1\relax
446   \fi
447   \ifx#1\eql@punct@tilde
448     \let#1\@empty
449   \fi
450 }
```

`\eqnpunct` Set the puncton 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:

```
451 \def\eqnpunct{%
452   \eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
453 \def\eql@punct@next@set#1{%
454   \ifmmode
455     \eql@punct@set\eql@punct@col{#1}%
456     \eql@punct@set\eql@punct@line{#1}%
457     \eql@punct@set\eql@punct@block{#1}%
458   \else
459     \eqnadopt{punct={#1}}%
460   \fi
461   \ignorespaces}
462 \def\eql@punct@next@setrelax{%
463   \ifmmode
464     \let\eql@punct@block\relax
465   \else
466     \eqnadopt{punct*}%
467   \fi
468   \ignorespaces}
```

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

```
469 \def\eql@punct@print@col{%
470   \ifx\eql@punct@col\@empty\else
471     \eql@punct@sep
472     \eql@punct@col
473   \fi
474 }
```

`\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:

```
475 \def\eql@punct@apply@col{%
476   \eql@punct@print@col
477   \let\eql@punct@col\@empty
478 }
```

Output the punctuation currently set for lines unless disabled:

`\eql@punct@print@line`

```
479 \def\eql@punct@print@line{%
480   \ifx\eql@punct@line\relax
481     \eql@punct@print@col
482   \else
483     \ifx\eql@punct@line\@empty\else
484       \eql@punct@sep
485       \eql@punct@line
486     \fi
487   \fi
488 }
```

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`

```
489 \def\eql@punct@apply@line{%
490   \ifx\eql@punct@line\relax
491   % \TODO hand down immediately?
492   \else
493     \ifx\eql@punct@line\@empty\else
494       \eql@punct@sep
495       \eql@punct@line
496     \fi
497     \let\eql@punct@line\relax
498     \let\eql@punct@col\@empty
499   \fi
500 }
```

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

```
501 \def\eql@punct@apply@block{%
502   \ifx\eql@punct@block\relax
503   % \TODO hand down immediately?
504   \else
505     \ifx\eql@punct@block\@empty\else
506       \eql@punct@sep
```

```

507     \eql@punct@block
508     \fi
509     \let\eql@punct@block\relax
510     \let\eql@punct@line\relax
511     \let\eql@punct@col\@empty
512     \fi
513 }

514 \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:

```

515 \def\eql@class@innerright@sel{%
516   \ifdim\eql@prevwidth=<=\z@
517     \eql@class@innerlead
518   \else
519     \eql@class@innerright
520   \fi
521 }

```

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

`\class@innerright@set`

`\@class@innerlead@set`

```

522 \def\eql@class@innerleft@set#1{%
523   \def\eql@class@innerleft{#1}%
524 }
525 \def\eql@class@innerright@set#1{%
526   \def\eql@class@innerright{#1}%
527   \let\eql@class@innerright@sel\eql@class@innerright
528 }
529 \def\eql@class@innerlead@set#1{%
530   \def\eql@class@innerlead{#1}%
531   \let\eql@class@innerright@sel\eql@class@innerright@sel@
532 }

```

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

`\eql@class@eqamp`

```

533 \def\eql@class@ampeq{%
534   \eql@class@innerleft@set{}%
535   \eql@class@innerright@set{{}}%
536 }
537 \def\eql@class@eqamp{%
538   \eql@class@innerleft@set{\mathrel{}}%
539   \eql@class@innerright@set{\mathrel{}}%
540   \eql@class@innerlead@set{{}}%
541 }
542 \eql@class@ampeq

```

4.3 Framed Cells

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

```

543 \let\eql@frame@cmd\undefined
544 \newdimen\eql@frame@margin@
545 \def\eql@frame@set[#1]{%
546   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
547 \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbox}
548 \def\eql@frame@measure{%
549   \setbox\z@\hbox{\eql@frame@cmd}}%
550   \eql@frame@margin@.5\wd\z@
551 }
552 \def\eql@frame@print{%
553   \setbox\eql@cellbox@\hbox{%
554     \eql@frame@cmd{\unhbox\eql@cellbox@}%
555   }%
556 }
557 \def\eql@frame@adjust{%
558   \setbox\eql@cellbox@\hbox{%
559     \eql@frame@cmd%
560     \unhbox\eql@cellbox@
561     \unkern
562     \unskip
563   }%
564   \hfil
565   \kern\z@
566 }%
567 }

```

4.4 Single-Line Composition

TODO: describe

`\eql@break@line`

```

568 \def\eql@break@line{\eql@srbgroup
569   \eql@ifnextgobble@tight~\eql@break@line@star
570   {\eql@punct@print@line\eql@break@line@star}}
571 \def\eql@break@line@star{%
572   \eql@ifstar@tight
573   {\eql@break@opt[\eql@break@line@shortsep]}%
574   {\eql@testopt@tight\eql@break@opt\eql@break@line@sep}}

```

`\eql@break@col`

```

575 \def\eql@break@col{\eql@srbgroup
576   \eql@ifnextgobble@tight~\eql@break@col@star
577   {\eql@punct@print@col\eql@break@col@star}}
578 \def\eql@break@col@star{%
579   \eql@ifstar@tight
580   {\eql@break@opt[\eql@break@col@shortsep]}%
581   {\eql@testopt@tight\eql@break@opt\eql@break@col@sep}}

```

`\eql@break@opt`

```

582 \def\eql@break@opt[#1]{\eql@sregroup\hspace{\glueexpr#1\relax}}

```

`\eql@break@join`

```

583 \def\eql@break@join{\eql@srbgroup
584   \eql@ifstar@tight

```

```

585     {\eql@break@join@opt[\eql@break@col@shortsep]}%
586     {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
587 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
588   \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

```

\eqnsep TODO: expand to lines and columns mode
\eqnbreak
\eqnjoin 589 \def\eqnsep{\eql@break@col}
          590 \def\eqnbreak{\eql@break@line}
          591 \def\eqnjoin{\eql@break@join}

```

4.5 Alternative Content Description

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

```

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

```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```

593 \let\eql@tags@autolabel\eql@false
594 \let\eql@tags@autotag\eql@true
595 \let\eql@tags@warn\eql@true

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

597 \let\eql@tagpos@doconvert\eql@false
598 \def\eql@tagpos@snap{4pt}

```

Registers.

```

599 \let\eql@numbering@mode@\undefined

600 \let\eql@numbering@active\eql@true
601 \let\eql@numbering@multi\eql@true

602 \let\eql@tags@container@\undefined
603 \def\eql@tags@container@clear{%
604   \let\eql@tags@label@\undefined
605   \let\eql@tags@name@\undefined
606   \let\eql@tags@tag@\undefined
607   \let\eql@tags@ref@\undefined
608   \let\eql@tags@anchor@\empty
609   \eql@tagpos@shift@z@
610   \eql@tagpos@smashup@z@
611   \eql@tagpos@smashdown@z@
612   \let\eql@tagpos@reserve\eql@true
613 }

```

```

614 \let\eq\@tags@label\@undefined
615 \let\eq\@tags@name\@undefined
616 \let\eq\@tags@tag\@undefined
617 \let\eq\@tags@ref\@undefined
618 \let\eq\@tags@frame@cmd\@firstofone

```

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

```

619 \newcount\eq\@tags@glabel@
620 \eq\@tags@glabel@\z@
621 \def\eq\@tags@glabel#equation.eql-\the\eq\@tags@glabel#}
622 \def\eq\@tags@glabel#stepf\global\advance\eq\@tags@glabel#\@ne}

623 \let\eq\@tagpos@continuous\eq\@false

624 \newcount\eq\@tagpos@row@
625 \newcount\eq\@tagpos@prevrow@
626 \newdimen\eq\@tagpos@shift@
627 \newdimen\eq\@tagpos@smashup@
628 \newdimen\eq\@tagpos@smashdown@
629 \newdimen\eq\@tagpos@current@
630 \newdimen\eq\@tagpos@plain@
631 \newdimen\eq\@tagpos@raised@
632 \newdimen\eq\@tagpos@target@
633 \newdimen\eq\@tagpos@headroom@
634 \newdimen\eq\@tagpos@footroom@

```

5.2 Schemes

TODO: describe

Table.

```

635 \def\eq\@numbering@tab@sub{sub}
636 \def\eq\@numbering@tab@all{all}
637 \def\eq\@numbering@tab@first{first}
638 \def\eq\@numbering@tab@last{last}
639 \def\eq\@numbering@tab@in{in}
640 \def\eq\@numbering@tab@out{out}
641 \def\eq\@numbering@tab@middle{middle}
642 \def\eq\@numbering@tab@best{best}
643 \def\eq\@numbering@tab@here{here}
644 \def\eq\@numbering@tab@top{top}
645 \def\eq\@numbering@tab@bottom{bottom}
646 \def\eq\@numbering@tab@center{center}
647 \def\eq\@numbering@tab@centerone{centerone}
648 \def\eq\@numbering@tab@median{median}
649 \def\eq\@numbering@tab@baseline{baseline}

650 \let\eq\@numbering@mode\eq\@numbering@tab@all
651 \let\eq\@numbering@mode@multi\eq\@numbering@tab@all
652 \let\eq\@numbering@mode@single\eq\@numbering@tab@out

```

TODO: describe

```

653 \let\eq\@numbering@tab@subeq\eq\@numbering@tab@sub
654 \let\eq\@numbering@tab@subequation\eq\@numbering@tab@sub
655 \let\eq\@numbering@tab@subequations\eq\@numbering@tab@sub

```

```

656 \let\eq@numbering@tab@mid\eq@numbering@tab@middle
657 \let\eq@numbering@tab@outside\eq@numbering@tab@out
658 \let\eq@numbering@tab@inside\eq@numbering@tab@in
659 \let\eq@numbering@tab@within\eq@numbering@tab@in
660 \let\eq@numbering@tab@opt\eq@numbering@tab@best
661 \let\eq@numbering@tab@optimal\eq@numbering@tab@best
662 \let\eq@numbering@tab@pick\eq@numbering@tab@here
663 \let\eq@numbering@tab@med\eq@numbering@tab@median
664 \eq@letcs{eq@numbering@tab@center*}\eq@numbering@tab@baseline
665 \eq@letcs{eq@numbering@tab@center!}\eq@numbering@tab@centerone

```

TODO: describe

```

666 \let\eq@numbering@tab@a\eq@numbering@tab@all
667 \let\eq@numbering@tab@s\eq@numbering@tab@sub
668 \let\eq@numbering@tab@f\eq@numbering@tab@first
669 \let\eq@numbering@tab@l\eq@numbering@tab@last
670 \let\eq@numbering@tab@o\eq@numbering@tab@out
671 \let\eq@numbering@tab@i\eq@numbering@tab@in
672 \let\eq@numbering@tab@h\eq@numbering@tab@here
673 \let\eq@numbering@tab@t\eq@numbering@tab@top
674 \let\eq@numbering@tab@b\eq@numbering@tab@bottom
675 \let\eq@numbering@tab@c\eq@numbering@tab@center
676 \let\eq@numbering@tab@m\eq@numbering@tab@median
677 \eq@letcs{eq@numbering@tab@+}\eq@numbering@tab@best
678 \eq@letcs{eq@numbering@tab@m*}\eq@numbering@tab@middle
679 \eq@letcs{eq@numbering@tab@c*}\eq@numbering@tab@baseline
680 \eq@letcs{eq@numbering@tab@c!}\eq@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

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

```

TODO: describe

```

682 \def\eq@numbering@init@sub{%
683   \let\eq@numbering@multi\eq@true
684   \ifdefined\eq@subequations@active
685     \let\eq@numbering@mode\eq@numbering@tab@all
686   \else
687     \let\eq@numbering@subeq@use\eq@true
688   \fi
689 }

690 \def\eq@numbering@init@first{\eq@tagpos@row@\@ne}
691 \def\eq@numbering@init@last{\eq@tagpos@row@\@MM}
692 \def\eq@numbering@init@here{\eq@tagpos@row@\m@ne}

```

TODO: describe

```

693 \def\eq@numbering@init@in{%
694   \ifdefined\eq@tagsleft
695     \eq@numbering@init@last
696   \else
697     \eq@numbering@init@first
698   \fi
699 }

```

TODO: describe

```

700 \def\eq@numbering@init@out{%

```



```

701 \ifdefined\eql@tagsleft
702   \eql@numbering@init@first
703 \else
704   \eql@numbering@init@last
705 \fi
706 }

```

TODO: describe

```

707 \def\eql@tagpos@eval@middle{%
708   \ifnum\eql@tagpos@row@=\z@
709     \eql@tagpos@row@=\numexpr(\eql@totalrows@
710       +\ifdefined\eql@tagsleft\z@\else\@ne\fi)/\tw@\relax
711   \fi
712 }

```

TODO: describe

```

713 \def\eql@tagpos@eval@best{%
714   \ifnum\eql@tagpos@row@=\z@
715     \let\eql@numbering@best@use\eql@true
716     \eql@numbering@init@out
717   \fi
718 }

```

TODO: describe

```

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

```

TODO: describe

```

720 \let\eql@numbering@init@top\eql@numbering@init@continuous
721 \def\eql@tagpos@eval@top{%
722   \eql@tagpos@current@\z@
723 }

```

TODO: describe

```

724 \let\eql@numbering@init@bottom\eql@numbering@init@continuous
725 \def\eql@tagpos@eval@bottom{%
726   \eql@tagpos@current@\dimexpr\eql@totalheight@
727     -\eql@tagheight@block@-\eql@tagdepth@block@\relax
728 }

```

TODO: describe

```

729 \let\eql@numbering@init@center\eql@numbering@init@continuous
730 \def\eql@tagpos@eval@center{%
731   \ifnum\eql@totalrows@=\@ne
732     \eql@tagpos@row@\@ne
733   \fi
734   \eql@tagpos@current@\dimexpr(\eql@totalheight@
735     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
736 }

```

TODO: describe

```

737 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
738 \def\eql@tagpos@eval@centerone{%
739   \eql@tagpos@current@\dimexpr(\eql@totalheight@
740     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
741 }

```

TODO: describe

```
742 \let\eq@numbering@init@baseline\eq@numbering@init@continuous
743 \def\eq@tagpos@eval@baseline{%
744   \eq@tagpos@current@\dimexpr(\eq@totalheight@
745     +\eq@topheight@-\eq@bottomdepth@)/\tw@-\eq@tagheight@block@\relax
746 }
```

TODO: describe

```
747 \let\eq@numbering@init@median\eq@numbering@init@continuous
748 \def\eq@tagpos@eval@median{%
749   \ifnum\eq@tagpos@row@=\z@
750     \ifodd\eq@totalrows@
751       \eq@tagpos@row@\numexpr(\eq@totalrows@+\@ne)/\tw@\relax
752     \else
753       \eq@tagpos@row@\numexpr(\eq@totalrows@+\tw@)/\tw@\relax
754       \eq@dimensions@get\eq@tagpos@row@
755       \advance\eq@tagpos@shift@\dimexpr\eq@line@height@
756         +(\eq@line@interline@-\eq@tagheight@block@
757           +\eq@tagdepth@block@)/\tw@\relax
758     \fi
759   \ifnum\eq@totalrows@=\@ne
760     \eq@tagpos@row@\@ne
761   \else
762     \eq@tagpos@adjust@eval@convert
763     \eq@tagpos@row@\z@
764   \fi
765 \fi
766 }
```

Selection.

```
767 \def\eq@numbering@set#1{%
768   \ifcsname eq@numbering@tab@#1\endcsname
769     \expandafter\let\expandafter\eq@numbering@mode
770     \csname eq@numbering@tab@#1\endcsname
771     \ifx\eq@numbering@mode\eq@numbering@tab@all
772       \let\eq@numbering@mode@multi\eq@numbering@mode
773     \else\ifx\eq@numbering@mode\eq@numbering@tab@sub
774       \let\eq@numbering@mode@multi\eq@numbering@mode
775     \else
776       \let\eq@numbering@mode@single\eq@numbering@mode
777     \fi\fi
778   \else
779     \eq@error{numbering mode '#1' unknown: setting mode to 'all'}%
780     \let\eq@numbering@mode\eq@numbering@tab@all
781   \fi
782 }
```

TODO: describe

```
783 \def\eq@numbering@init{%
784   \let\eq@numbering@multi\eq@false
785   \let\eq@tagpos@continuous\eq@false
786   \let\eq@numbering@subeq@use\eq@false
787   \let\eq@numbering@best@use\eq@false
788   \eq@tagpos@row@\z@
789   \csname eq@numbering@init@\eq@numbering@mode\endcsname
790   \ifdefined\eq@numbering@active
```

```

791 \let\eq@numbering@eqnswinit\@eqnswtrue
792 \else
793 \let\eq@numbering@eqnswinit\@eqnswfalse
794 \fi
795 \let\eq@numbering@active\eq@false
796 }

```

5.3 Interface

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

```

797 \eq@amsmath@after{
798 \let\eq@print@eqnum@default\print@eqnum
799 \let\eq@incr@eqnum@default\incr@eqnum
800 }

```

TODO: describe

```

801 \protected\def\donumber{%
802 \if@eqnsw\else
803 \global\@eqnswtrue
804 \ifx\print@eqn\@empty
805 \global\let\print@eqn\eq@print@eqnum@default
806 \fi
807 \ifx\incr@eqn\@empty
808 \global\let\incr@eqn\eq@incr@eqnum@default
809 \fi
810 \fi
811 }

```

TODO: reconsider operation

`\numberhere`

```

812 \protected\def\eq@numberhere{%
813 \ifdefined\eq@numbering@multi
814 \global\@eqnswtrue
815 \else
816 \global\eq@tagpos@row@\eq@row@
817 \fi
818 }

```

TODO: describe

`\numbernext`

```

819 \protected\def\eq@numbernext{%
820 \ifdefined\eq@numbering@multi
821 \global\@eqnswfalse
822 \else
823 \ifdefined\eq@tagpos@continuous\else
824 \ifnum\eq@tagpos@row@=\eq@row@
825 \global\advance\eq@tagpos@row@\@ne
826 \fi
827 \fi
828 \fi
829 }

```

Activation Trigger.

```
830 \def\eql@tags@autoenable{%
831   \global\@eqnswtrue
832   \ifnum\eql@tagpos@row@=\m@ne
833     \numberhere
834   \fi
835 }
```

Labels. **TODO:** describe

\eql@label@org

```
836 \let\eql@label@org\label
```

TODO: describe

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

TODO: describe

```
838 \protected\def\eql@label{%
839   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
840 }
```

TODO: describe

```
841 \def\eql@tags@add@labelname[#1]#2{%
842   \def\eql@tmp{#1}%
843   \ifx\eql@tmp\eql@testopt@default\else
844     \eql@tags@add@name{#1}%
845   \fi
846   \eql@tags@add@label{#2}%
847 }
```

TODO: describe

```
848 \def\eql@tags@set@label#1{%
849   \ifdefined\eql@tags@warn
850     \ifdefined\eql@tags@label
851       \eql@warn@label@multiple{#1}%
852     \fi
853   \fi
854   \def\eql@tags@label{#1}%
855 }
```

TODO: describe

```
856 \def\eql@tags@set@name#1{%
857   \ifdefined\eql@tags@warn
858     \ifdefined\eql@tags@name
859       \eql@warn@name@multiple
860     \fi
861   \fi
862   \def\eql@tags@name{#1}%
863 }
```

TODO: describe

```
864 \def\eql@tags@add@label#1{%
865   \ifdefined\eql@tags@autolabel
866     \eql@tags@autoenable
```

```

867 \fi
868 \global\eqL@appendexpand\eqL@tags@container{%
869   \noexpand\eqL@tags@set@label{#1}}%
870 }

```

TODO: describe

```

871 \def\eqL@tags@add@name#1{%
872   \protected@edef\eqL@tmp{\noexpand\eqL@tags@set@name{#1}}%
873   \global\eqL@appendmacro\eqL@tags@container\eqL@tmp
874 }

```

TODO: describe

```

875 \def\eqL@tags@addblock@label#1{%
876   \eqL@appendexpand\eqL@tags@container@block{%
877     \noexpand\eqL@tags@set@label{#1}}%
878 }

```

TODO: describe

```

879 \def\eqL@tags@addblock@name#1{%
880   \protected@edef\eqL@tmp{\noexpand\eqL@tags@set@name{#1}}%
881   \eqL@appendmacro\eqL@tags@container@block\eqL@tmp
882 }

```

Tags. **TODO:** describe

`\eqL@tag@default`

```

883 \protected\def\eqL@tag@default{%
884   \eqL@warn@here\tag
885   \eqL@tag@gobble
886 }
887 \let\tag\eqL@tag@default

```

`\eqL@tag@gobble`

```

888 \def\eqL@tag@gobble{%
889   \eqL@ampprotecttwo\eqL@teststaropt@tight\eqL@gobbleoptone\eqL@gobbleoptone{}}

```

TODO: describe

```

890 \protected\def\eqL@tag{%
891   \eqL@ampprotecttwo\eqL@teststaropt@tight
892   {\eqL@tags@add@tagform@off\eqL@tags@add@tagref}{\eqL@tags@add@tagref}
893   \eqL@testopt@default
894 }

```

`\eqL@tags@add@tagref`

```

895 \def\eqL@tags@add@tagref[#1]#2{%
896   \def\eqL@tmp{#1}%
897   \ifx\eqL@tmp\eqL@testopt@default\else
898     \eqL@tags@add@ref{#1}%
899   \fi
900   \eqL@tags@add@tag{#2}%
901 }

```

TODO: describe

```

902 \def\eql@tags@set@tag#1{%
903   \ifdefined\eql@tags@warn
904     \ifdefined\eql@tags@tag
905       \eql@warn@tag@multiple
906     \fi
907   \fi
908 \def\eql@tags@tag{#1}%
909 }

```

TODO: describe

```

910 \def\eql@tags@set@ref#1{%
911   \ifdefined\eql@tags@warn
912     \ifdefined\eql@tags@ref
913       \eql@warn@ref@multiple
914     \fi
915   \fi
916 \def\eql@tags@ref{#1}%
917 }

```

TODO: describe

```

918 \def\eql@tags@add@tag#1{%
919   \ifdefined\eql@tags@autotag
920     \eql@tags@autoenable
921   \fi
922 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
923 \global\eql@appendmacro\eql@tags@container\eql@tmp
924 }

```

TODO: describe

```

925 \def\eql@tags@add@ref#1{%
926 \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
927 \global\eql@appendmacro\eql@tags@container\eql@tmp
928 }

```

tags@add@tagform@off

```

929 \def\eql@tags@add@tagform@off{%
930 \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
931 }

```

TODO: describe

```

932 \def\eql@tags@addblock@tag#1{%
933 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
934 \eql@appendmacro\eql@tags@container@block\eql@tmp
935 }

```

TODO: describe

```

936 \def\eql@tags@addblock@ref#1{%
937 \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
938 \eql@appendmacro\eql@tags@container@block\eql@tmp
939 }

```

TODO: describe

```

940 \def\eql@tags@addblock@tagform@off{%
941 \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
942 }

```

Raise Tags.

`\raisetag`

```
943 \def\eql@raisetag@default{%
944   \eql@warn@here\raisetag
945   \eql@raisetag@gobble
946 }

947 \def\eql@raisetag@gobble{%
948   \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble
949 }
```

TODO: describe

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

951 \def\eql@raisetag{%
952   \eql@ampprotecttwo\eql@ifstar@tight\eql@tags@add@raiseshift\eql@raisetag@test
953 }

954 \def\eql@raisetag@test#1{%
955   \def\eql@tmpa{#1}\def\eql@tmpb{!}%
956   \ifx\eql@tmpa\eql@tmpb
957     \eql@tags@add@forceraise
958   \else
959     \eql@tags@add@raisesmash{#1}%
960   \fi
961 }

962 \def\eql@tags@add@raiseshift#1{%
963   \global\eql@appendexpand\eql@tags@container{%
964     \advance\eql@tagpos@shift@the\glueexpr#1\relax\relax}%
965 }

966 \def\eql@tags@add@raisesmash#1{%
967   \dimen@glueexpr#1\relax
968   \ifdim\dimen@<\z@
969     \global\eql@appendexpand\eql@tags@container{%
970       \advance\eql@tagpos@smashdown@the\dimen@\relax}%
971   \else
972     \global\eql@appendexpand\eql@tags@container{%
973       \advance\eql@tagpos@smashup@the\dimen@\relax}%
974   \fi
975 }

976 \def\eql@tags@add@forceraise{%
977   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
978 }
```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```
979 \def\eql@numbering@settools{%
980   \let\label\eql@label
981   \let\tag\eql@tag
```

```

982 \let\raisetag\eql@raisetag
983 \let\numberhere\eql@numberhere
984 \let\numbernext\eql@numbernext
985 }

```

TODO: not necessary anymore

```

986 \def\eql@numbering@settools@gobble{%
987 \let\label\eql@label@gobble
988 \let>tag\eql>tag@gobble
989 \let\raisetag\eql@raisetag@gobble
990 \let\numberhere\relax
991 \let\numbernext\relax
992 }

```

```

993 \def\eql@numbering@autoblock{%
994 \begingroup
995 \let\eql@tags@warn\eql@false
996 \eql@tags@container@block
997 \ifdefined\eql@tags@autolabel
998 \ifdefined\eql@tags@label
999 \global\@eqnswtrue
1000 \fi
1001 \fi
1002 \ifdefined\eql@tags@autotag
1003 \ifdefined\eql@tags@tag
1004 \global\@eqnswtrue
1005 \fi
1006 \fi
1007 \endgroup
1008 }

```

```

1009 \def\eql@numbering@warnunused{%
1010 \ifdefined\eql@tags@label
1011 \eql@warn@label@unused
1012 \fi
1013 \ifdefined\eql@tags@name
1014 \eql@warn@name@unused
1015 \fi
1016 \ifdefined\eql@tags@tag
1017 \eql@warn@tag@unused
1018 \fi
1019 \ifdefined\eql@tags@erf
1020 \eql@warn@ref@unused
1021 \fi
1022 }

```

Single Line. TODO: describe

```

1023 \def\eql@numbering@single@init{%
1024 \let\eql@numbering@multi\eql@false
1025 \eql@numbering@settools
1026 \eql@numbering@eqnswinit
1027 \eql@numbering@autoblock
1028 \global\let\eql@tags@container\eql@tags@container@block
1029 \let\eql@tags@warn\eql@true
1030 }

```

```

1031 \def\eql@numbering@single@eval{%

```



```

1032 \ifnum\eql@tagpos@row@=\m@ne
1033   \@eqnswfalse
1034 \fi
1035 }

```

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

```

1036 \def\eql@numbering@measure@init{%
1037   \eql@numbering@settools
1038   \ifdefined\eql@numbering@multi\else
1039     \eql@numbering@eqnswinit
1040     \eql@numbering@autoblock
1041   \fi
1042   \global\let\eql@tags@container\eql@tags@container@block
1043   \let\eql@tags@warn\eql@true
1044 }

```

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

```

1045 \def\eql@numbering@measure@line@begin{%
1046   \ifdefined\eql@numbering@multi
1047     \global\eql@numbering@eqnswinit
1048   \fi
1049 }

```

TODO: describe

```

1050 \def\eql@numbering@measure@blocktag{%
1051   \ifdefined\eql@numbering@multi
1052     \@eqnswfalse
1053   \else
1054     \ifnum\eql@tagpos@row@=\m@ne
1055       \@eqnswfalse
1056     \fi
1057     \ifnum\eql@totalrows@=\z@
1058       \@eqnswfalse
1059     \fi
1060   \fi
1061 }

```

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!

```

1062 \def\eql@numbering@print@init{%
1063   \let\eql@tags@warn\eql@false
1064   \ifdefined\eql@numbering@multi
1065     \eql@numbering@settools
1066     \global\let\eql@tags@container\eql@tags@container@block
1067   \else
1068     \let\eql@tags@container@block\eql@tags@container
1069     \eql@numbering@settools@gobble
1070   \fi
1071 }

```

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

```

1072 \def\eql@numbering@print@block@begin{%
1073   \ifdefined\eql@numbering@multi\else

```

```

1074 \ifnum\eql@tagpos@row@>\z@
1075 \eql@tags@makeblockanchor
1076 \global\eql@appendexpand\eql@tags@container@block{%
1077 \def\noexpand\eql@tags@anchor{%
1078 \unexpanded\expandafter{\eql@tags@anchor}}}%
1079 \fi
1080 \fi
1081 \ifdefined\eql@numbering@subeq@use
1082 \eql@tags@printsubeqlabel
1083 \fi
1084 }

```

TODO: describe

```

1085 \def\eql@numbering@print@line@begin{%
1086 \ifdefined\eql@numbering@multi
1087 \global\eql@numbering@eqnswinit
1088 \fi
1089 }

```

TODO: describe

```

1090 \def\eql@numbering@print@line@eval{%
1091 \ifdefined\eql@numbering@multi
1092 \if@eqnsw
1093 \eql@tags@container
1094 \fi
1095 \else
1096 \ifnum\eql@tagpos@row@=\eql@row@
1097 \@eqnswtrue
1098 \eql@tags@container@block
1099 \else
1100 \@eqnswfalse
1101 \fi
1102 \fi
1103 }

```

5.5 Positioning

TODO: describe

```

1104 \def\eql@tagpos@single@eval{%
1105 \if@eqnsw
1106 \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1107 \ifnum\eql@tagpos@row@>\@ne
1108 \eql@tagpos@row@\@ne
1109 \fi
1110 \ifdefined\eql@tagpos@doconvert
1111 \let\eql@tagpos@continuous\eql@true
1112 \fi
1113 \ifdefined\eql@tagpos@continuous
1114 \eql@tagpos@single@eval@continuous
1115 \fi
1116 \else
1117 \eql@tagpos@row@\z@
1118 \fi
1119 \eql@tagpos@prevrow@\z@
1120 \eql@tagpos@headroom@\z@
1121 \eql@tagpos@footroom@\z@

```

1122 }

TODO: describe

```
1123 \def\eq\tagpos@single@eval@continuous{%
1124 \ifnum\eq\tagpos@row@>\z@
1125 \eq\tagpos@target@\eq\tagpos@shift@
1126 \else
1127 \eq\tagpos@target@\dimexpr\eq\line@height@
1128 -\eq\tagpos@current@+\eq\tagpos@shift@-\eq\tagheight@block@\relax
1129 \fi
1130 \eq\tagpos@row@\@ne
1131 \ifdim\ifdim\eq\tagpos@target@<\z@-\fi
1132 \eq\tagpos@target@<\glueexpr\eq\tagpos@snap\relax
1133 \eq\tagpos@target@\z@
1134 \fi
1135 }
```

TODO: describe

```
1136 \def\eq\tagpos@adjust@eval{%
1137 \ifeqsw
1138 \csname eq\tagpos@eval@\eq\numbering@mode\endcsname
1139 \ifnum\eq\tagpos@row@>\eq\totalrows@
1140 \eq\tagpos@row@\eq\totalrows@
1141 \fi
1142 \ifdefined\eq\tagpos@doconvert
1143 \let\eq\tagpos@continuous\eq>true
1144 \fi
1145 \ifdefined\eq\tagpos@continuous
1146 \ifnum\eq\tagpos@row@>\z@
1147 \eq\tagpos@adjust@eval@convert
1148 \fi
1149 \eq\tagpos@adjust@eval@continuous
1150 \fi
1151 \else
1152 \eq\tagpos@row@\z@
1153 \eq\tagpos@prevrow@\z@
1154 \fi
1155 }
```

TODO: describe

```
1156 \def\eq\tagpos@adjust@eval@convert{%
1157 \eq\tagpos@current@\z@
1158 \eq\dimensions@for{%
1159 \ifnum\eq\row@<\eq\tagpos@row@
1160 \advance\eq\tagpos@current@\dimexpr\eq\line@interline@
1161 +\eq\line@height@+\eq\line@depth@\relax
1162 \fi
1163 \ifnum\eq\row@=\eq\tagpos@row@
1164 \advance\eq\tagpos@current@\dimexpr\eq\line@interline@
1165 +\eq\line@height@-\eq\tagheight@block@\relax
1166 \fi
1167 }%
1168 }
```

TODO: describe

```
1169 \def\eq\tagpos@adjust@eval@continuous{%
1170 \dimen@\dimexpr\eq\tagpos@current@-\eq\tagpos@shift@\relax
```

```

1171 \eql@tagpos@row@\eql@totalrows@
1172 \eql@tagpos@prevrow@\z@
1173 \eql@tagpos@headroom@\z@
1174 \eql@tagpos@footroom@\z@
1175 \eql@dimensions@for{%
1176   \ifnum\eql@tagpos@row=\eql@totalrows@
1177     \eql@tagpos@headroom@\eql@line@interline@
1178     \eql@tagpos@target@\dimexpr\eql@line@interline@
1179       +\eql@line@height@-\dimen@-\eql@tagheight@block@relax
1180   \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1181     \eql@tagpos@target@<\glueexpr\eql@tagpos@snap@relax
1182     \advance\dimen@\eql@tagpos@target@
1183     \eql@tagpos@target@\z@
1184   \fi
1185   \ifdim\eql@tagpos@target@>%
1186     \ifdefined\eql@tagsleft-1sp@relax\else\z@\fi
1187     \eql@tagpos@row@\eql@row@
1188     \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne@relax
1189   \fi
1190   \advance\dimen@-\dimexpr\eql@line@interline@
1191     +\eql@line@depth@+\eql@line@height@relax
1192   \fi
1193   \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne@relax
1194     \eql@tagpos@footroom@\eql@line@interline@
1195   \fi
1196 }%
1197 }

```

TODO: describe

```

1198 \def\eql@tagpos@print@line@eval{%
1199   \ifdefined\eql@tagpos@continuous
1200     \eql@tagpos@print@line@eval@continuous
1201   \else
1202     \eql@tagpos@print@line@eval@discrete
1203   \fi
1204 }

```

TODO: describe

```

1205 \def\eql@tagpos@print@line@eval@continuous{%
1206   \if@eqnsw
1207     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@relax
1208     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@relax
1209     \eql@tagpos@plain@\eql@tagpos@target@
1210     \@tempdima\dimexpr\eql@line@height@+\eql@tagpos@headroom@
1211       -\ht\eql@tagbox@relax
1212     \@tempdimb\dimexpr-\eql@line@depth@-\eql@tagpos@footroom@
1213       +\dp\eql@tagbox@relax
1214     \ifnum\eql@row@=\@ne
1215       \@tempdima.5\maxdimen
1216     \fi
1217     \ifnum\eql@row@=\eql@totalrows@
1218       \@tempdimb-.5\maxdimen
1219     \fi
1220     \ifdim\eql@tagpos@plain@>\@tempdima
1221       \ifdim\eql@tagpos@plain@>\@tempdimb
1222         \ifdim\@tempdima>\@tempdimb
1223           \eql@tagpos@plain@\@tempdima
1224         \else

```

```

1225     \eql@tagpos@plain@\@tempdimb
1226   \fi
1227   \fi
1228   \else
1229     \ifdim\eql@tagpos@plain@<\@tempdimb
1230     \ifdim\@tempdima>\@tempdimb
1231       \eql@tagpos@plain@\@tempdimb
1232     \else
1233       \eql@tagpos@plain@\@tempdima
1234     \fi
1235   \fi
1236 \fi
1237 \ifnum\eql@tagpos@prevrow>\z@
1238   \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1239   \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1240     \eql@tagpos@raised@\eql@tagpos@plain@
1241     \let\eql@tagpos@reserve\eql@false
1242   \fi
1243 \else
1244   \ifdim\eql@tagpos@target@>%
1245     \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1246     \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1247     \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1248       \eql@tagpos@raised@\eql@tagpos@plain@
1249       \let\eql@tagpos@reserve\eql@false
1250     \fi
1251   \else
1252     \eql@tagpos@raised@\dimexpr-\eql@line@depth@
1253     -\ht\eql@tagbox@\relax
1254     \ifdim\eql@tagpos@raised@<\eql@tagpos@plain@\else
1255       \eql@tagpos@raised@\eql@tagpos@plain@
1256       \let\eql@tagpos@reserve\eql@false
1257     \fi
1258   \fi
1259 \fi
1260 \else
1261   \ifnum\eql@tagpos@prevrow@=\eql@row@
1262     \eql@tagwidth@\eql@tagwidth@block@
1263   \else
1264     \let\eql@tagpos@reserve\eql@false
1265   \fi
1266 \fi
1267 }

```

TODO: describe

```

1268 \def\eql@tagpos@print@line@eval@discrete{%
1269   \if@eqnsw
1270     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1271     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1272     \eql@tagpos@plain@\eql@tagpos@shift@
1273     \eql@tagpos@headroom@\z@
1274     \eql@tagpos@footroom@\z@
1275     \ifdim\eql@tagpos@shift@>%
1276       \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1277     \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1278   \else
1279     \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1280   \fi

```

```

1281 \else
1282   \let\eql@tagpos@reserve\eql@false
1283 \fi
1284 }

```

TODO: describe

```

1285 \def\eql@tagpos@print@line@end{%
1286   \ifdefined\eql@tagpos@continuous
1287     \ifnum\eql@tagpos@prevrow@=\eql@row@
1288       \ifdefined\eql@tagpos@reserve
1289         \global\eql@appendexpand\eql@tags@container@block{%
1290           \advance\eql@tagpos@headroom@ \the\dimexpr\eql@line@height@
1291             +\eql@line@depth@\relax\relax}%
1292         \eql@displaybreak@star\M
1293       \fi
1294     \fi
1295 \fi
1296 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1297 \let\eql@SK@loaded\eql@false
1298 \let\eql@SK@label\@gobble
1299 \let\eql@SK@clearlabel\@empty
1300 \let\eql@SK@setlabel\@gobble
1301 \let\eql@SK@printlabel@right\@empty
1302 \let\eql@SK@printlabel@left\@empty
1303 \let\eql@SK@printlabel@line\@empty
1304 \def\eql@label@clean{\eql@label@org}
1305 \AddToHook{package/showkeys/after}{
1306   \let\eql@SK@loaded\eql@true
1307   \def\eql@SK@label#1{\SK@\SK@@label#1}
1308   \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1309   \eql@SK@clearlabel
1310   \def\eql@SK@@label#1>#2\SK@{%
1311     \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1312   }
1313   \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1314   \def\eql@SK@printlabel@right{%
1315     \ifx\eql@SK@lab\relax\else
1316       \rlap{\kern\marginparsep\eql@SK@lab}%
1317     \eql@SK@clearlabel
1318     \fi
1319   }
1320   \def\eql@SK@printlabel@left{%
1321     \ifx\eql@SK@lab\relax\else
1322       \llap{\eql@SK@lab\kern\marginparsep}%
1323     \eql@SK@clearlabel
1324     \fi
1325   }
1326   \def\eql@SK@printlabel@line{%
1327     \ifx\eql@SK@lab\relax\else
1328       \dimen@\prevdepth
1329       \nointerlineskip
1330       \ifdefined\eql@tagsleft

```

```

1331     \llap{%
1332         \eql@SK@lab
1333         \kern\marginparsep
1334     }%
1335     \eql@SK@clearlabel
1336 \else
1337     \rlap{%
1338         \dimen@\displaywidth
1339         \advance\dimen@\marginparsep
1340         \kern\dimen@
1341         \eql@SK@lab
1342     }%
1343 \fi
1344 \eql@SK@clearlabel
1345 \prevdepth\dimen@
1346 \fi
1347 }
1348 \let\eql@label@org\label
1349 \def\eql@label@clean{\let\SK@\gobbletwo\eql@label@org}
1350 }

```

Labels.

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

```

1351 \def\eql@composetag@label{%
1352     \eql@SK@clearlabel
1353     \ifdefined\eql@tags@label
1354         \eql@SK@setlabel\eql@tags@label
1355         \ifdefined\eql@tags@name
1356             \let\@currentlabelname\eql@tags@name
1357         \else
1358             \let\@currentlabelname\eql@tags@name@generic
1359         \fi
1360     \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1361 \fi
1362 }

```

TODO: describe

```

1363 \def\eql@tags@printsubeqlabel{%
1364     \eql@tags@container@parent
1365     \ifdefined\eql@tags@label
1366         \eql@tags@makeblockanchor
1367         \eql@SK@setlabel\eql@tags@label
1368     \begingroup
1369         \def\@currentcounter{equation}%
1370         \eql@tags@anchor
1371         \let\@currentlabelname\eql@tags@name@generic
1372         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1373         \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1374     \endgroup
1375     \eql@SK@printlabel@line
1376 \fi
1377 }

```

Hyperref Anchors. **TODO:** describe

```

1378 \let\eqL@Hy@anchor@gobble
1379 \AddToHook{package/hyperref/after}{
1380   \def\eqL@Hy@anchor#1{%
1381     \Hy@raisedlink{\hyper@anchor{#1}}%
1382   }%
1383 }

```

TODO: describe

```

1384 \def\eqL@tags@makeblockanchor{%
1385   \eqL@tags@glabel@step
1386   \eqL@Hy@anchor\eqL@tags@glabel
1387   \edef\eqL@tags@anchor{%
1388     \def\noexpand\thepage{\thepage}%
1389     \def\noexpand\@currentHref{\eqL@tags@glabel}%
1390   }%
1391 }

```

TODO: describe

qL@composetag@anchor

```

1392 \def\eqL@composetag@anchor{%
1393   \ifdefined\eqL@tags@tag
1394     \def\@currentcounter{equation}%
1395     \ifdefined\eqL@tags@ref
1396       \let\@currentlabel\eqL@tags@ref
1397     \else
1398       \protected@edef\@currentlabel{\p@equation\eqL@tags@tag}%
1399     \fi
1400     \eqL@tags@glabel@step
1401     \edef\@currentHref{\eqL@tags@glabel}%
1402     \eqL@Hy@anchor\@currentHref
1403   \else
1404     \refstepcounter{equation}%
1405     \protected@edef\eqL@tags@tag{\theequation}%
1406   \fi
1407   \eqL@tags@anchor
1408 }

```

Tag Layout. **TODO:** describe

```

1409 \def\eqL@tags@taglayout@set@direct#1{%
1410   \def\eqL@tags@taglayout##1{#1}%
1411 }
1412 \def\eqL@tags@taglayout@set#1{%
1413   \def\eqL@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1414 }

```

TODO: describe

```

1415 \def\eqL@tags@tagform@set@direct#1{%
1416   \def\eqL@tags@tagform##1{#1}%
1417 }
1418 \def\eqL@tags@tagform@set#1#2#3{%
1419   \def\eqL@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1420 }

1421 \eqL@tags@taglayout@set{#1}

```



```

1422 \eql@tags@tagform@set({#1})
1423 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1424 \protected\def\tagform{\eql@tags@tagform}
1425 \protected\def\tagbox{\eql@tags@taglayout}
1426 \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:

```

1427 \protected\def\eqleqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1428 \def\eql@composetag@tag{%
1429   \eql@tagging@tagbegin
1430   \eql@tags@frame@cmd{%
1431     \eql@tags@taglayout{%
1432       \eql@tags@tagform\eql@tags@tag
1433       \eql@tagging@tagsave
1434     }%
1435   }%
1436   \eql@tagging@tagend
1437 }

```

5.7 Tag Composition

TODO: describe

```

1438 \def\eql@composetag@measure{%
1439   \ifdefined\eql@tags@tag\else
1440     \stepcounter{equation}%
1441     \let\eql@tags@tag\theequation
1442   \fi
1443   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1444   \ifdefined\eql@numbering@multi
1445     \global\let\eql@tags@container\eql@tags@container@clear
1446   \fi
1447 }

```

TODO: describe

```

1448 \def\eql@composetag@print{%
1449   \eql@composetag@anchor
1450   \eql@composetag@label
1451   \ifdefined\eql@tagsleft
1452     \eql@SK@printlabel@left
1453     \eql@composetag@tag
1454   \else
1455     \eql@composetag@tag
1456     \eql@SK@printlabel@right
1457   \fi
1458   \global\let\eql@tags@container\eql@tags@container@clear
1459 }

```

TODO: describe

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

```

1460 \def\eql@tagbox@make#1{%

```

```

1461 \setbox\eq@tagbox@hbox{\eq@strut@tag@lign#1}%
1462 \eq@tagwidth@wd\eq@tagbox@
1463 \ifdim\eq@tagwidth@<\eq@tagwidthmin@
1464 \eq@tagwidth@\eq@tagwidthmin@
1465 \fi
1466 \advance\eq@tagwidth@\eq@tagsepmin@
1467 }

```

TODO: describe

```

1468 \def\eq@tagbox@print@adjustheadroom{%
1469 \dimen@dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1470 \ifdim\dimen@>\z@
1471 \ifdim\dimen@>\eq@tagpos@headroom@
1472 \ht\eq@tagbox@dimexpr\ht\eq@tagbox@-\eq@tagpos@headroom@\relax
1473 \else
1474 \ht\eq@tagbox@dimexpr\eq@line@height@-\eq@tagpos@current@\relax
1475 \fi
1476 \fi
1477 }

```

TODO: describe

```

1478 \def\eq@tagbox@print@adjustfootroom{%
1479 \dimen@dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax
1480 \ifdim\dimen@>\z@
1481 \ifdim\dimen@>\eq@tagpos@footroom@
1482 \dp\eq@tagbox@dimexpr\dp\eq@tagbox@-\eq@tagpos@footroom@\relax
1483 \else
1484 \dp\eq@tagbox@dimexpr\eq@line@depth@+\eq@tagpos@current@\relax
1485 \fi
1486 \fi
1487 }

```

TODO: describe

```

1488 \def\eq@tagbox@print@extendabove{%
1489 \dimen@dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1490 \ifdim\dimen@>\z@
1491 \global\eq@appendexpand\eq@display@container{%
1492 \eq@display@aboveextend@the\dimen@\relax}%
1493 \fi
1494 }

```

TODO: describe

```

1495 \def\eq@tagbox@print@extendbelow{%
1496 \dimen@dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax
1497 \ifdim\dimen@>\z@
1498 \global\eq@appendexpand\eq@display@container{%
1499 \eq@display@belowextend@the\dimexpr\dimen@\relax}%
1500 \fi
1501 }

```

TODO: describe

```

1502 \def\eq@tagbox@print@prepare{%
1503 \ifdefined\eq@tagpos@reserve
1504 \eq@tagpos@current@\eq@tagpos@plain@
1505 \else
1506 \eq@tagpos@current@\eq@tagpos@raised@
1507 \fi

```

```

1508 \ifdim\eql@tagpos@headroom@>\z@
1509   \eql@tagbox@print@adjustheadroom
1510 \fi
1511 \ifdim\eql@tagpos@footroom@>\z@
1512   \eql@tagbox@print@adjustfootroom
1513 \fi
1514 \ifnum\eql@row@=\@ne
1515   \eql@tagbox@print@extendabove
1516 \fi
1517 \ifnum\eql@row@=\eql@totalrows@
1518   \eql@tagbox@print@extendbelow
1519 \fi
1520 }

```

TODO: describe

```

1521 \def\eql@tagbox@print@tagsright{%
1522   \eql@tagbox@print@prepare
1523   \kern-\wd\eql@tagbox@
1524   \raise\eql@tagpos@current@\box\eql@tagbox@
1525 }

```

TODO: describe

```

1526 \def\eql@tagbox@print@tagsleft{%
1527   \eql@display@firstavail@set\z@
1528   \eql@tagbox@print@prepare
1529   \wd\eql@tagbox@\z@
1530   \raise\eql@tagpos@current@\box\eql@tagbox@
1531 }

```

ql@tagbox@print@cell

```

1532 \def\eql@tagbox@print@cell{%
1533   \eql@tagging@tagaddbox
1534   \ifdefined\eql@tagsleft
1535     \ifdefined\eql@tagpos@reserve
1536       \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1537         \let\eql@tagpos@reserve\eql@false
1538       \fi
1539     \fi
1540     \if@eqnsw
1541       \eql@tagbox@print@tagsleft
1542     \fi
1543     \kern\displaywidth
1544   \else
1545     \kern\displaywidth
1546     \ifdefined\eql@tagpos@reserve
1547       \ifdim\eql@tagwidth@>%
1548         \dimexpr\displaywidth-\eql@line@width@+\eql@tagfuzz@\relax
1549         \let\eql@tagpos@reserve\eql@false
1550       \fi
1551     \fi
1552     \if@eqnsw
1553       \eql@tagbox@print@tagsright
1554     \fi
1555   \fi
1556 }

```

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:

```
1557 \eql@amsmath@undefine\c@parentequation
1558 \eql@amsmath@undefine\theparentequation
1559 \ifdefined\c@parentequation\else
1560 \newcounter{parentequation}
1561 \fi
```

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

```
1562 \def\eql@subequations@template{\theparentequation\alph{equation}}
```

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

```
1563 \let\eql@subequations@active\eql@false
```

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

```
1564 \def\eql@subequations@init{%
1565   \edef\eql@subequations@restorecounter{%
1566     \global\c@equation\the\c@equation\relax}%
1567   \eql@tags@container@block
1568   \ifdefined\eql@tags@tag
1569     \eql@tags@glabel@step
1570     \protected@edef\theHparentequation{\eql@tags@glabel}%
1571     \protected@edef\theparentequation{\eql@tags@tag}%
1572   \else
1573     \advance\c@equation\@ne
1574     \protected@edef\theparentequation{\theequation}%
1575     \ifdefined\theHequation
1576       \protected@edef\theHparentequation{\theHequation}%
1577     \fi
1578   \fi
1579   \global\c@parentequation\c@equation
1580   \global\c@equation\z@
1581   \let\theequation\eql@subequations@template
1582   \def\theHequation{\theHparentequation.\arabic{equation}}}%
1583 }
```

`@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:

```

1584 \def\eq@subequations@close{%
1585   \ifnum\c@equation=\z@
1586     \eq@subequations@restorecounter
1587   \else
1588     \global\c@equation\c@parentequation
1589   \fi
1590 }

```

6.2 Environment

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

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

```

1591 \def\eq@subequations@start{%
1592   \let\eq@tags@container@block\eq@tags@container@clear
1593   \eq@nextopt@process{subequations}%
1594   \eq@subequations@init
1595   \eq@tags@glabel@step
1596   \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1597   \eq@Hy@anchor\eq@subequations@currentHref
1598   \edef\eq@subequations@thepage{\thepage}%
1599   \def\@currentcounter{equation}%
1600   \let\@currentHref\eq@subequations@currentHref
1601   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1602   \eq@tags@container@block
1603   \ifdefined\eq@tags@name
1604     \let\@currentlabelname\eq@tags@name
1605   \else
1606     \let\@currentlabelname\eq@tags@name@generic
1607   \fi
1608   \let\eq@subequations@active\eq@true
1609   \ifdefined\eq@tags@label
1610     \eq@SK@label\eq@tags@label
1611   \fi
1612   \ignorespaces
1613 }

```

`\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:

```

1614 \def\eq@subequations@end{%
1615   \ifnum\c@equation>\z@
1616     \eq@tags@container@block
1617     \ifdefined\eq@tags@label
1618       \begingroup
1619         \def\@currentcounter{equation}%
1620         \let\thepage\eq@subequations@thepage
1621         \let\@currentHref\eq@subequations@currentHref
1622 % \TODO how about tag* ?! also within equations!
1623         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1624         \ifdefined\eq@tags@name
1625           \let\@currentlabelname\eq@tags@name
1626         \else
1627           \let\@currentlabelname\eq@tags@name@generic

```

```

1628     \fi
1629     \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1630   \endgroup
1631   \fi
1632   \fi
1633   \eq@subequations@close
1634   \ignorespacesafterend
1635 }

```

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

```

1636 \newenvironment{eq@subequations}{%
1637 (dev)\eq@dev@enterenv
1638   \eq@subequations@testall\eq@subequations@start%
1639 }{%
1640   \eq@subequations@end
1641 (dev)\eq@dev@leaveenv
1642 }

```

TODO: describe

```

1643 \def\eq@subequations@testall{\eq@parseopt\eq@subequations@parseopt}
1644 \def\eq@subequations@parseopt{%
1645   \ifx\eq@parseopt@token[%]
1646     \let\eq@parseopt@next\eq@parseopt@opt
1647     \fi
1648   \ifx\eq@parseopt@token\eq@atxi
1649     \let\eq@parseopt@next\eq@parseopt@label
1650     \fi
1651   \ifx\eq@parseopt@token\eq@atxii
1652     \let\eq@parseopt@next\eq@parseopt@label
1653     \fi
1654   \ifx\eq@parseopt@token\label
1655     \let\eq@parseopt@next\eq@parseopt@end
1656     \fi
1657 }

```

6.3 Subequation Scheme

TODO: describe

```

1658 \def\eq@numbering@subeq@init{%
1659   \let\eq@save@theequation\theequation
1660   \let\eq@save@theHequation\theHequation
1661   \eq@subequations@init
1662   \let\eq@tags@container@parent\eq@tags@container@block
1663   \let\eq@tags@container@block\eq@tags@container@clear
1664 }

```

TODO: describe

```

1665 \def\eq@numbering@subeq@test{%
1666   \ifnum\eq@tagrows@<\tw@
1667     \let\eq@tags@container@block\eq@tags@container@parent
1668     \let\eq@numbering@subeq@use\eq@false
1669     \let\theequation\eq@save@theequation
1670     \let\theHequation\eq@save@theHequation
1671     \eq@subequations@restorecounter

```

```
1672 \fi
1673 }
```

TODO: describe

```
1674 % \TODO note must not use setcounter here (when calc is loaded)
1675 \def\eq@numbering@subeq@close{%
1676 \eq@subequations@close
1677 }
```

7 Display Equations Support

TODO: describe

```
1678 \let\eq@display@injectbefore\@undefined
1679 \let\eq@display@injectafter\@undefined
1680 \let\eq@interline@container\@undefined
1681 \def\eq@interline@container@clear{%
1682 \eq@displaybreak@pen@\@MM
1683 \eq@vspaceskip\z@skip
1684 }
```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1685 \interdisplaylinepenalty\@M
```

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

```
1686 \def\eq@getdsp@pen#1{%
1687 \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1688 }
```

TODO: allow a displaybreak before equations

```
1689 \protected\def\eq@displaybreak@default{%
1690 \eq@warning{Invalid use of \string\displaybreak}{}}%
1691 \eq@teststaroropt@loose\@gobble\eq@gobbleopt{}}
1692 \eq@amsmath@after{\let\eq@displaybreak@default\displaybreak}
1693 \eq@amsmath@let\displaybreak\eq@displaybreak@default
```

```
1694 \newcount\eq@displaybreak@pen@
1695 \newcount\eq@displaybreak@prepen@
1696 \newcount\eq@displaybreak@postpen@
```

TODO: describe

```
1697 \protected\def\eq@displaybreak{%
1698 \relax
1699 \eq@ampprotecttwo\eq@teststaroropt@tight
1700 \eq@displaybreak@star\eq@displaybreak@level{4}}%
1701 }
```

```
1702 \def\eq@displaybreak@star#1{%
1703 \global\eq@appendexpand\eq@interline@container{%
```

```
1704 \eql@displaybreak@open@the\numexpr#1\relax\relax}%
1705 }
```

```
1706 \def\eql@displaybreak@level[#1]{%
1707 \ifnum#1<\z@
1708 \global\eql@append\eql@interline@container{\eql@displaybreak@open@\@MM}%
1709 \else
1710 \global\eql@appendexpand\eql@interline@container{%
1711 \eql@displaybreak@open@-\@getpen{#1}\relax}%
1712 \fi
1713 }
```

TODO: describe

```
1714 \def\eql@displaybreak@pre#1{%
1715 \ifnum#1<\z@
1716 \eql@displaybreak@prepen@\@MM
1717 \else
1718 \eql@displaybreak@prepen@-\@getpen{#1}\relax
1719 \fi
1720 }
```

TODO: describe

```
1721 \def\eql@displaybreak@post#1{%
1722 \ifnum#1<\z@
1723 \eql@displaybreak@postpen@\@MM
1724 \else
1725 \eql@displaybreak@postpen@-\@getpen{#1}\relax
1726 \fi
1727 }
```

TODO: describe

```
1728 \def\eql@displaybreak@inter#1{%
1729 \ifnum#1<\z@
1730 \interdisplaylinepenalty\@M
1731 \else
1732 \interdisplaylinepenalty\eql@getdsp@open{#1}\relax
1733 \fi
1734 }
```

7.2 Explicit Vertical Space

TODO: describe

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

```
1735 \newskip\eql@vspaceskip@

1736 \let\eql@vspace@org\vspace
1737 \def\eql@vspace{%
1738 \ifvmode
1739 \expandafter\eql@vspace@immediate
1740 \else
1741 \expandafter\eql@vspace@line
1742 \fi
1743 }
```


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

```

1744 \def\eql@vspace@line{%
1745   \eql@ifstar@loose\eql@vspace@addfixedbefore\eql@vspace@add
1746 }
1747 \def\eql@vspace@add#1{%
1748   \global\eql@appendexpand\eql@interline@container{%
1749     \advance\eql@vspaceskip@the\glueexpr#1\relax\relax}}
1750 \def\eql@vspace@addfixedbefore#1{%
1751   \global\eql@appendexpand\eql@interline@container{%
1752     \noexpand\eql@append\noexpand\eql@display@injectbefore{%
1753       \skip@the\glueexpr#1\relax\relax
1754       \penalty\@M
1755       \vskip\skip@
1756       \global\advance\eql@line@interline@\skip@
1757     }%
1758   }%
1759 }
1760 \def\eql@vspace@addfixedafter#1{%
1761   \global\eql@appendexpand\eql@interline@container{%
1762     \noexpand\eql@append\noexpand\eql@display@injectafter{%
1763       \dimen@\prevdepth
1764       \hrule\@height\z@
1765       \skip@the\glueexpr#1\relax\relax
1766       \penalty\@M
1767       \vskip\skip@
1768       \global\advance\eql@line@interline@\skip@
1769       \prevdepth\dimen@
1770     }%
1771   }%
1772 }

```

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

```

1773 \def\eql@vspace@immediate{%
1774   \noalign\bgroup
1775   \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
1776 }
1777 \def\eql@vspace@fixed#1{%
1778   \skip@\glueexpr#1\relax
1779   \ifnum\eql@row@=\@ne
1780     \global\eql@appendexpand\eql@display@container{%
1781       \advance\eql@abovespace@the\skip@\relax}%
1782   \else\ifnum\eql@row@>\eql@totalrows@
1783     \global\eql@appendexpand\eql@display@container{%
1784       \advance\eql@belowspace@the\skip@\relax}%
1785   \else
1786     \dimen@\prevdepth
1787     \hrule\@height\z@
1788     \penalty\@M
1789     \vskip\skip@
1790     \global\advance\eql@line@interline@\skip@
1791     \prevdepth\dimen@
1792   \fi\fi
1793   \egroup
1794 }
1795 \def\eql@vspace@discardable#1{%
1796   \skip@\glueexpr#1\relax

```

```

1797 \ifnum\eqL@row@=\@ne
1798 \global\eqL@appendexpand\eqL@display@container{%
1799 \advance\eqL@abovespace@the\skip@\relax}%
1800 \else\ifnum\eqL@row@>\eqL@totalrows@
1801 \global\eqL@appendexpand\eqL@display@container{%
1802 \advance\eqL@belowspace@the\skip@\relax}%
1803 \else
1804 \vskip\skip@
1805 \global\advance\eqL@line@interline@\skip@
1806 \fi\fi
1807 \egroup
1808 }

```

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.

```

1809 \newbox\eqL@strutbox@
1810 \def\eqL@strut@depth{.3}
1811 \def\eqL@strut{\copy\eqL@strutbox@}
1812 \let\eqL@strut@cell\eqL@strut
1813 \let\eqL@strut@tag\eqL@strut
1814 \def\eqL@strut@make{%
1815 \setbox\eqL@strutbox@\hbox{%
1816 \tempdimb\dimexpr
1817 \eqL@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1818 \tempdima\dimexpr
1819 \normalbaselineskip-\normallineskiplimit-\tempdimb\relax
1820 \vrule\@height\tempdima\@depth\tempdimb\@width\z@
1821 }
1822 }
1823 \AtBeginDocument{\eqL@strut@make}

```

TODO: describe

```

1824 \def\eqL@spread@set{%
1825 \eqL@spread@\dimexpr\glueexpr\eqL@spread@val\relax
1826 +\normalbaselineskip-\baselineskip\relax
1827 \ifdim\eqL@spread@>\z@
1828 \openup\eqL@spread@
1829 \ifdefined\spread@equation
1830 \let\spread@equation\@empty
1831 \fi
1832 \fi
1833 }

```

7.4 Entry and Exit

TODO: describe

```

1834 \let\eqL@beamerbasecolor@fix\@empty
1835 \AddToHook{package/beamerbasecolor/after}{%
1836 \def\eqL@beamerbasecolor@fix{%
1837 \donotcolorouterdisplaymaths

```

```

1838 \donotcoloroutermaths
1839 \beamer@setdisplaymathcolor
1840 }%
1841 }

```

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

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

```

1842 \newskip\eql@abovespace@
1843 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

1844 \def\eql@display@enter{%
1845 \ifnoskipsec\leavevmode\par\fi
1846 \ifvmode
1847 \eql@prevdepth@\prevdepth
1848 \nointerlineskip
1849 \noindent
1850 \else
1851 \eql@prevdepth@\maxdimen
1852 \fi
1853 \eql@beamerbasecolor@fix
1854 }

```

`\eql@display@adjust`

```

1855 \def\eql@display@adjust{%
1856 \ifdefined\eql@display@linewidth
1857 \displaywidth\glueexpr\eql@display@linewidth\relax
1858 \advance\displaywidth-\displayindent
1859 \fi
1860 \ifdefined\eql@display@marginleft
1861 \advance\displaywidth\displayindent
1862 \displayindent\glueexpr\eql@display@marginleft\relax
1863 \advance\displaywidth-\displayindent
1864 \fi
1865 \ifdefined\eql@display@marginright
1866 \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1867 \fi
1868 \ifdim\displaywidth<\z@
1869 \displaywidth\z@
1870 \fi
1871 }

```

`\eql@display@init`

```

1872 \def\eql@display@init{%
1873 \let\displaybreak\eql@displaybreak
1874 \let\eql@vspace@org\vspace
1875 \let\vspace\eql@vspace
1876 \let\eqncontrol\eql@control
1877 \let\eql@display@injectbefore\@empty
1878 \let\eql@display@injectafter\@empty
1879 \eql@spread@set
1880 \eql@strut@make
1881 \let\eql@frame@cmd\@undefined
1882 }

```

`\eql@display@print`

```
1883 \def\eql@display@print{%
1884   \let\eql@display@container\@empty
1885   \eql@display@firstavail@z@
1886   \eql@display@aboveextend@z@
1887   \eql@display@belowextend@z@
1888   \global\let\eql@interline@container\eql@interline@container@clear
1889 }
```

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

```
1890 \def\eql@display@halign@init#1{%
1891   \eql@row@z@
1892   \eql@prevgraf@\prevgraf
1893   \everycr{\noalign{%
1894     \global\advance\eql@row@\@ne
1895     \prevgraf\numexpr\prevgraf+\@ne\relax
1896     #1%
1897   }}%
1898 }
```

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

```
1899 \def\eql@display@halign@start{%
1900   \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1901   \ifdim\eql@prevdepth@=\maxdimen\else
1902     \prevdepth\eql@prevdepth@
1903   \fi
1904   \ifdim\prevdepth=-\@m\p\else
1905     \ifdefined\eql@display@height
1906       \skip@\baselineskip
1907       \advance\skip@-\glueexpr\eql@display@height\relax
1908       \advance\skip@-\prevdepth\relax
1909       \ifdim\skip@<\lineskiplimit
1910         \skip@\lineskip
1911       \fi
1912       \advance\skip@-\eql@spread@\relax
1913       \vskip\skip@
1914       \nointerlineskip
1915     \else
1916       \vskip-\eql@spread@\relax
1917     \fi
1918   \fi
1919 }
```

TODO: describe

```
1920 \def\eql@display@vspace{%
1921   \advance\abovedisplayskip\eql@abovespace@
1922   \advance\belowdisplayskip\eql@belowspace@
1923 }
```

TODO: describe

```
1924 \def\eql@display@vspace@native{%
1925   \advance\abovedisplayskip\eql@abovespace@
1926   \advance\belowdisplayskip\eql@belowspace@
1927   \advance\abovedisplayshortskip\eql@abovespace@
1928   \advance\belowdisplayshortskip\eql@belowspace@
1929 }
```

TODO: describe

```
1930 \def\eqldisplay@penalty{%
1931   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1932     \postdisplaypenalty\eqldisplaybreak@postpen@
1933   \fi
1934   \ifnum\eqldisplaybreak@open@=\@MM\else
1935     \postdisplaypenalty\eqldisplaybreak@open@
1936   \fi
1937   \ifnum\eqldisplaybreak@prepen@=\@MM\else
1938     \predisplaypenalty\eqldisplaybreak@prepen@
1939   \fi
1940 }
```

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

```
1941 \def\eqldisplay@halign@end{%
1942   \eql@interline@container
1943   \eqldisplay@injectbefore
1944   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
1945   \ifdefined\eqldisplay@depth
1946     \prevdepth\glueexpr\eqldisplay@depth\relax
1947   \fi
1948 }
```

`\eqldisplay@close` **TODO:** there seems to be an offset of 1em in `preplaysize` 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?

```
1949 \def\eqldisplay@close{%
1950   \eqldisplay@container
1951   \ifdim\eqldisplay@firstavail@<\z@
1952     \eqldisplay@firstavail@\z@
1953   \fi
1954   \eql@skip@mode@leave@\z@
1955   \ifdim\eql@prevdepth@=\maxdimen
1956     \ifdim\preplaysize=-\maxdimen
1957       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1958       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1959     \else
1960       \eql@skip@mode@above@\z@
1961       \eql@skip@mode@below@\z@
1962       \advance\eqldisplay@firstavail@\displayindent
1963       \ifdim\eqldisplay@firstavail@>\preplaysize
1964         \ifcase\eql@skip@mode@short\relax
1965           \or
1966             \eql@skip@mode@above@\@ne
1967           \or
1968             \eql@skip@mode@above@\@ne
1969             \ifnum\eql@totalrows@=\@ne
1970               \eql@skip@mode@below@\@ne
1971             \fi
1972           \or
1973             \eql@skip@mode@above@\@ne
1974             \eql@skip@mode@below@\@ne
1975           \fi
1976         \fi
1977       \fi
1978     \else
1979       \ifdim\eql@prevdepth@=-\@m\p@
```

```

1980     \eql@skip@mode@above@\eql@skip@mode@top@above\relax
1981     \eql@skip@mode@below@\eql@skip@mode@top@below\relax
1982     \else
1983     \eql@skip@mode@above@\eql@skip@mode@par@above\relax
1984     \eql@skip@mode@below@\eql@skip@mode@par@below\relax
1985     \fi
1986     \fi
1987     \ifcase\eql@skip@mode@above@
1988     \or\or\or
1989     \predisplaypenalty\z@
1990     \or
1991     \predisplaypenalty\z@
1992     \fi
1993     \ifcase\eql@skip@mode@below@
1994     \or\or\or
1995     \eql@skip@mode@leave@\@ne
1996     \or
1997     \eql@skip@mode@leave@\tw@
1998     \fi
1999     \ifdefined\eql@skip@force@above
2000     \eql@skip@mode@above@\eql@skip@force@above\relax
2001     \fi
2002     \ifdefined\eql@skip@force@below
2003     \eql@skip@mode@below@\eql@skip@force@below\relax
2004     \fi
2005     \ifdefined\eql@skip@force@leave
2006     \eql@skip@mode@leave@\eql@skip@force@leave\relax
2007     \fi
2008     \ifnum\eql@skip@mode@leave@>\z@
2009     \postdisplaypenalty\z@
2010     \fi
2011     \ifcase\eql@skip@mode@above@
2012     \abovedisplayskip\glueexpr\eql@skip@long@above\relax
2013     \or
2014     \abovedisplayskip\glueexpr\eql@skip@short@above\relax
2015     \or
2016     \abovedisplayskip\glueexpr\eql@skip@cont@above\relax
2017     \or
2018     \abovedisplayskip\glueexpr\eql@skip@par@above\relax
2019     \or
2020     \abovedisplayskip\glueexpr\eql@skip@top@above\relax
2021     \or
2022     \abovedisplayskip\z@skip
2023     \or
2024     \abovedisplayskip\glueexpr\eql@skip@med@above\relax
2025     \or
2026     \abovedisplayskip\glueexpr\eql@skip@custom@above\relax
2027     \fi
2028     \ifcase\eql@skip@mode@below@
2029     \belowdisplayskip\glueexpr\eql@skip@long@below\relax
2030     \or
2031     \belowdisplayskip\glueexpr\eql@skip@short@below\relax
2032     \or
2033     \belowdisplayskip\glueexpr\eql@skip@cont@below\relax
2034     \or
2035     \belowdisplayskip\glueexpr\eql@skip@par@below\relax
2036     \or
2037     \belowdisplayskip\glueexpr\eql@skip@top@below\relax

```

```

2038 \or
2039   \belowdisplayskip\z@skip
2040 \or
2041   \belowdisplayskip\glueexpr\eq@skip@med@below\relax
2042 \or
2043   \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
2044 \fi
2045 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
2046 \eq@interline@container
2047 \advance\eq@belowspace@\eq@vspaceskip@
2048 \eq@display@penalty
2049 \eq@display@vspace
2050 \skip@\glueexpr\eq@skip@tag@above\relax
2051 \ifdim\skip@>\abovedisplayskip
2052   \skip@\abovedisplayskip
2053 \fi
2054 \advance\abovedisplayskip-\eq@display@aboveextend@\relax
2055 \ifdim\abovedisplayskip<\skip@
2056   \abovedisplayskip\skip@
2057 \fi
2058 \skip@\glueexpr\eq@skip@tag@below\relax
2059 \ifdim\skip@>\belowdisplayskip
2060   \skip@\belowdisplayskip
2061 \fi
2062 \ifdim\eq@display@belowextend@>\z@
2063   \advance\belowdisplayskip-\eq@display@belowextend@\relax
2064   \ifdim\belowdisplayskip<\skip@
2065     \belowdisplayskip\skip@
2066   \fi
2067 \fi
2068 }

```

TODO: describe

```

2069 \def\eq@display@leave{%
2070   \prevgraf\eq@prevgraf@
2071   \ifcase\eq@skip@mode@leave@
2072     \or
2073     \endgraf
2074   \or
2075     \endgraf
2076     \prevdepth-\@m\p@
2077   \fi
2078 }

```

TODO: describe

```

2079 \def\eq@display@nest{%
2080   \let\displaybreak\eq@displaybreak@default
2081   \let\intertext\eq@intertext@default
2082   \let\vspace\eq@vspace@org
2083 }

```

TODO: describe

```

2084 \def\eq@display@restore{%
2085   \let\label\eq@label@org
2086   \let>tag\eq@tag@default
2087   \let\raisetag\eq@raisetag@default
2088   \let\displaybreak\eq@displaybreak@default
2089   \let\intertext\eq@intertext@default

```

```
2090 \let\vspace\eq@vspace@org
2091 }
```

TODO: describe

```
2092 \eq@append\@arrayparboxrestore{%
2093 \eq@display@restore
2094 \ifdefined\eq@ampproof@active
2095 \eq@amprevert
2096 \fi
2097 \@displayfalse
2098 }
```

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)

```
2099 \def\eq@stack@enable{%
2100 \let\eq@stack@save@equations\eq@stack@save@equations@
2101 \let\eq@stack@save@box\eq@stack@save@box@
2102 }
```

TODO: describe

```
2103 \let\eq@stack@save@equations\eq@stack@enable
2104 \let\eq@stack@save@box\eq@stack@enable
2105 \let\eq@stack@restore\@empty
```

TODO: describe

```
2106 \def\eq@stack@save@reg#1{\global#1\the#1\relax}
2107 \def\eq@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

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

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

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

```
2108 \def\eq@stack@save@equations@{%
2109 \let\eq@stack@numbering@eqnswinit\eq@numbering@eqnswinit
2110 \let\eq@stack@cell@container\eq@cell@container
2111 \let\eq@stack@tags@container\eq@tags@container
2112 \let\eq@stack@interline@container\eq@interline@container
2113 \let\eq@stack@block@container\eq@display@container
2114 \let\eq@stack@dimensions@tab\eq@dimensions@tab
2115 \edef\eq@stack@restore{%
2116 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2117 \eq@stack@save@let\eq@stack@numbering@eqnswinit\eq@numbering@eqnswinit
2118 \eq@stack@save@let\eq@stack@cell@container\eq@cell@container
2119 \eq@stack@save@let\eq@stack@tags@container\eq@tags@container
2120 \eq@stack@save@let\eq@stack@interline@container\eq@interline@container
2121 \eq@stack@save@let\eq@stack@dimensions@tab\eq@dimensions@tab
2122 \eq@stack@save@let\eq@stack@block@container\eq@display@container
2123 \eq@stack@save@reg\eq@column@
2124 \eq@stack@save@reg\eq@totalcolumns@
2125 \eq@stack@save@reg\eq@line@avail@
2126 \eq@stack@save@reg\eq@line@pos@
2127 \eq@stack@save@reg\eq@line@width@
```



```

2128 \eql@stack@save@reg\eql@line@depth@
2129 \eql@stack@save@reg\eql@line@height@
2130 \eql@stack@save@reg\eql@line@prevdepth@
2131 \eql@stack@save@reg\eql@line@interline@
2132 \eql@stack@save@reg\eql@totalheight@
2133 \eql@stack@save@reg\eql@tagwidth@max@
2134 \eql@stack@save@reg\eql@tagpos@row@
2135 \eql@stack@save@reg\eql@row@
2136 \eql@stack@save@reg\eql@tagrows@
2137 }%
2138 }

```

TODO: describe

```

2139 \def\eql@stack@save@box@{%
2140 \let\eql@stack@cell@container\eql@cell@container
2141 \edef\eql@stack@restore{%
2142 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2143 \eql@stack@save@reg\eql@row@
2144 }%
2145 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2146 \def\eql@measure@init#1#2{%
2147 \eql@dimensions@reset
2148 \let\eql@display@container\@empty
2149 \eql@numbering@measure@init
2150 \eql@row@z@
2151 \eql@totalheight@z@
2152 \eql@totalrows@M
2153 \eql@line@prevdepth@-@m@p@
2154 \eql@line@interline@z@
2155 \tabskipz@skip
2156 \everycrf\@noalign{%
2157 \global\advance\eql@row@\@ne
2158 #1%
2159 }%
2160 \global\let\eql@interline@container\eql@interline@container@clear
2161 \eql@measure@savestate
2162 \eql@display@halign@letcr{#2}%
2163 }

```

TODO: describe

```

2164 \def\eql@measure@tag{%
2165 \eql@tagwidth@z@
2166 \ifdefined\eql@numbering@multi
2167 \if@eqnsw
2168 \eql@tags@container
2169 \eql@tagbox@make\eql@composetag@measure

```

```

2170     \ifdefined\eq\tagpos@reserve\else
2171     \eq\tagwidth@z@
2172     \fi
2173     \fi
2174     \fi
2175 }

```

TODO: describe

```

2176 \def\eq\measure@endrow{%
2177   \ifdim\eq\line@prevdepth@=-\m\p@\else
2178     \dimen@\dimexpr\baselineskip-\eq\line@height@-\eq\line@prevdepth@\relax
2179     \ifdim\dimen@<\lineskiplimit
2180       \dimen@\lineskip
2181       \fi
2182     \advance\eq\line@interline@\dimen@
2183     \fi
2184     \eq\dimensions@endrow
2185     \ifdim\eq\tagwidth@>\eq\tagwidth@max@
2186       \global\eq\tagwidth@max@\eq\tagwidth@
2187       \fi
2188     \ifdim\eq\tagwidth@>z@
2189       \global\advance\eq\tagrows@\@ne
2190       \fi
2191     \global\advance\eq\totalheight@\dimexpr
2192       \eq\line@interline@+\eq\line@height@+\eq\line@depth@
2193     \global\eq\line@interline@z@
2194     \global\eq\line@prevdepth@\eq\line@depth@
2195 }

```

TODO: describe

```

2196 \def\eq\measure@close{%
2197   \advance\eq\row@-\tw@
2198   \eq\totalrows@\eq\row@
2199   \ifnum\eq\totalrows@>z@
2200     \eq\dimensions@get@\@ne
2201     \eq\topheight@\dimexpr\eq\line@height@+\eq\line@interline@\relax
2202     \eq\dimensions@get\eq\totalrows@
2203     \eq\bottomdepth@\eq\line@depth@
2204     \fi
2205     \eq\numbering@measure@blocktag
2206     \begingroup
2207     \eq\tags@container
2208     \if@eqnsw
2209       \eq\tagbox@make\eq\composetag@measure
2210       \ifdefined\eq\tagpos@reserve\else
2211         \eq\tagwidth@z@
2212         \fi
2213       \eq\dimensions@saveblocktag
2214     \else
2215       \eq\dimensions@savenoblocktag
2216       \eq\numbering@warnunused
2217     \fi
2218     \endgroup
2219     \eq\dimensions@getz@
2220     \eq\measure@restorestate
2221 }

```

```

measure@restorestate
\q@measure@savestate
2222 \let\eq@measure@restorestate\@empty
2223 \def\eq@measure@savestate{%
2224   \begingroup
2225     \def\@elt##1{%
2226       \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2227     \global\edef\@gtempa{\cl@ckpt}%
2228   \endgroup
2229   \let\eq@measure@restorestate\@gtempa
2230 }

```

8.2 Line Breaks

TODO: describe

`\eq@display@cr`

```

2231 \protected\def\eq@display@cr{\eq@srbgroup
2232   \eq@ifnextchar@tight~%
2233     {\numbernext\let\eq@punct@line\@empty\eq@display@cr@star}%
2234     \eq@display@cr@star
2235 }

```

`\eq@display@cr@star`

```

2236 \def\eq@display@cr@star{%
2237   \eq@teststaropt@tight{%
2238     \global\eq@append\eq@interline@container{\eq@displaybreak@pen@MM}%
2239     \eq@display@cr@opt}%
2240   \eq@display@cr@opt\z@skip
2241 }

```

`\eq@display@cr@opt`

```

2242 \def\eq@display@cr@opt[#1]{\eq@sregroup
2243   \eq@display@endline
2244   \cr
2245   \noalign{%
2246     \eq@interline@container
2247     \eq@display@injectbefore
2248     \ifnum\eq@displaybreak@pen@MM
2249       \penalty\interdisplaylinepenalty
2250     \else
2251       \penalty\eq@displaybreak@pen@
2252     \fi
2253     \advance\eq@vspaceskip@glueexpr#1\relax
2254     \vskip\eq@vspaceskip@
2255     \global\advance\eq@line@interline@\eq@vspaceskip@
2256     \eq@display@injectafter
2257     \global\let\eq@interline@container\eq@interline@container@clear
2258   }%
2259 }

```

`\display@halign@letcr`

```

2260 \def\eq@display@halign@letcr#1{%

```

```

2261 \let\\eql@display@cr
2262 \let\eql@display@endline#1%
2263 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2264 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2265 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2266 \def\eql@intertext@process{%
2267   \eql@display@endline
2268   \cr
2269   \ifmeasuring@
2270     \expandafter\@gobble
2271   \else
2272     \expandafter\eql@intertext@print
2273   \fi
2274 }

```

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

```

2275 \def\eql@intertext@print#1{%
2276   \noalign{%
2277     \eql@display@halign@end
2278     \let\eql@skip@force@below\z@
2279     \let\eql@skip@force@above\z@
2280     \eql@setkeys{intertext}\eql@intertext@opt
2281     \openup-\eql@spread@
2282     \penalty\postdisplaypenalty
2283     \ifcase\eql@skip@force@below\relax
2284       \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2285     \or
2286       \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2287     \or
2288       \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2289     \or
2290       \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2291     \or
2292       \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2293     \or
2294       \advance\eql@vspaceskip@\z@skip
2295     \or
2296       \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2297     \or
2298       \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2299     \fi
2300     \vskip\eql@vspaceskip@
2301     \global\let\eql@interline@container\eql@interline@container@clear
2302     \vbox{%
2303       \@parboxrestore
2304       \ifdim
2305         \ifdim@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2306       \else

```

```

2307     \parshape\@ne
2308     \@totalleftmargin\linewidth
2309     \fi
2310     \noindent
2311     \prevgraf\eq\@prevgraf@
2312     \ignorespaces
2313     #1%
2314     \par
2315     \global\eq\@prevgraf@\prevgraf
2316   }%
2317   \penalty\predisplaypenalty
2318   \ifcase\eq\@skip@force@above\relax
2319     \vskip\glueexpr\eq\@skip@long@above\relax
2320   \or
2321     \vskip\glueexpr\eq\@skip@short@above\relax
2322   \or
2323     \vskip\glueexpr\eq\@skip@cont@above\relax
2324   \or
2325     \vskip\glueexpr\eq\@skip@par@above\relax
2326   \or
2327     \vskip\glueexpr\eq\@skip@top@above\relax
2328   \or
2329     \vskip\z@skip
2330   \or
2331     \vskip\glueexpr\eq\@skip@med@above\relax
2332   \or
2333     \vskip\glueexpr\eq\@skip@custom@above\relax
2334   \fi
2335 %   \eq\@prevdepth@\maxdimen
2336   \eq\@prevdepth@\z@
2337   \eq\@display@\halign@start
2338 }
2339 }

```

TODO: describe

```

2340 \newenvironment{eq\@intertext}{%
2341   \eq\@testopt@tight\eq\@intertext@{}%
2342 }{%
2343   \aftergroup\eq\@intertext@after
2344   \ignorespacesafterend
2345 }

```

TODO: describe

```

2346 \def\eq\@intertext@env{intertext}
2347 \def\eq\@intertext@[#1]{%
2348   \global\def\eq\@intertext@opt{#1}%
2349   \ifx\@currenvir\eq\@intertext@env
2350     \expandafter\eq\@scan@env\expandafter\eq\@intertext@inject
2351   \else
2352     \expandafter\eq\@intertext@process
2353   \fi
2354 }

```

TODO: describe

```

2355 \def\eq\@intertext@inject{%
2356   \global\edef\eq\@intertext@after{%
2357     \noexpand\eq\@intertext@process{%
2358       \ifx\eq\@scan@body\eq\@scan@body@dump

```

```

2359     \eql@scan@body@dump
2360     \else
2361     \noexpand\scantokens{\eql@scan@body@dump}%
2362     \fi
2363   }%
2364 }%
2365 }

```

8.4 Line Marks

TODO: describe

```

2366 \def\eql@markline@pos@below{below}
2367 \def\eql@markline@pos@bottom{bottom}
2368 \def\eql@markline@pos@baseline{baseline}
2369 \let\eql@markline@pos\eql@markline@pos@baseline
2370 \let\eql@markline@shift\z@
2371 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2372 \def\eql@markline@symbol{}

```

TODO: describe

```

2373 \def\eql@markline@select#1{%
2374   \let\eql@markline@shift\z@
2375   \eql@setkeys{markline}{#1}%
2376   \eql@markline@print
2377 }

```

TODO: describe

```

2378 \def\eql@markline@print{%
2379   \dimen@\

```

TODO: describe

```

2402 \def\eql@markline@inject#1{%
2403   \let\eql@markline@push\eql@false

```

```

2404 \ifx\eq\markline@pos\eq\markline@pos@below\else
2405 \ifdefined\eq\tagleft\else
2406 \ifx\eq\equations@main\eq\multi@main
2407 \ifdefined\eq\numbering@multi
2408 \if@eqnsw
2409 \let\eq\markline@push\eq>true
2410 \fi
2411 \else
2412 \ifnum\eq\row@=\eq\tagpos@row@
2413 \let\eq\markline@push\eq>true
2414 \fi
2415 \fi
2416 \else
2417 \if@eqnsw
2418 \let\eq\markline@push\eq>true
2419 \fi
2420 \fi
2421 \fi
2422 \fi
2423 \ifdefined\eq\markline@push
2424 \global\eq\append\eq\interline@container{%
2425 \eq\append\eq\display@injectbefore{\eq\markline@select{push,#1}}}%
2426 \else
2427 \global\eq\append\eq\interline@container{%
2428 \eq\append\eq\display@injectbefore{\eq\markline@select{#1}}}%
2429 \fi
2430 }

```

TODO: describe

```

2431 \def\eq\markline@amsthm@opt[#1]{\eq\markline@inject{qed,#1}}
2432 \def\eq\markline@amsthm@staropt[#1]{\eq\markline@inject{qed,push,#1}}
2433 \def\eq\markline@amsthm@qed{\eq\teststaropt@tight
2434 \eq\markline@amsthm@staropt\eq\markline@amsthm@opt{}}
2435 \def\eq\markline@amsthm@register#1{\eq\letcs{#1@qed}\eq\markline@amsthm@qed}

```

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:

```

2436 \newcount\eq\shape@pos@
2437 \newdimen\eq\shape@amount@
2438 \let\eq\shape@lastrow\eq>false

```

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

```

\eq\marginleft@min@ (dimen)
\eq\marginright@ (dimen)
\eq\centeroffset@ (dimen)
2439 \newdimen\eq\marginleft@
2440 \newdimen\eq\marginright@
2441 \newdimen\eq\marginleft@min@
2442 \newdimen\eq\centeroffset@

```

9.2 Shape Schemes

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

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

```
2443 \def\eql@shape@tab@default{default}
2444 \def\eql@shape@tab@left{left}
2445 \def\eql@shape@tab@center{center}
2446 \def\eql@shape@tab@right{right}
2447 \def\eql@shape@tab@first{first}
2448 \def\eql@shape@tab@hanging{hanging}
2449 \def\eql@shape@tab@steps{steps}
```

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

```
2450 \let\eql@shape@tab@def\eql@shape@tab@default
2451 \let\eql@shape@tab@\eql@shape@tab@default
2452 \let\eql@shape@tab@l\eql@shape@tab@left
2453 \let\eql@shape@tab@c\eql@shape@tab@center
2454 \let\eql@shape@tab@r\eql@shape@tab@right
2455 \let\eql@shape@tab@rc\eql@shape@tab@first
2456 \let\eql@shape@tab@indent\eql@shape@tab@first
2457 \let\eql@shape@tab@hang\eql@shape@tab@hanging
2458 \let\eql@shape@tab@lc\eql@shape@tab@hanging
2459 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
2460 \let\eql@shape@tab@lcr\eql@shape@tab@steps
```

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

```
2461 \let\eql@shape@mode\eql@shape@tab@default
```

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

```
2462 \def\eql@shape@set#1{%
2463   \ifcsname eql@shape@tab@#1\endcsname
2464     \expandafter\let\expandafter\eql@shape@mode
2465     \csname eql@shape@tab@#1\endcsname
2466   \else
2467     \eql@error{shape '#1' unknown: setting to default}%
2468     \let\eql@shape@mode\eql@shape@tab@default
2469   \fi
2470 }
```

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

```
2471 \def\eql@shape@layoutcenter@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2472 \def\eql@shape@layoutcenter@center{\eql@shape@pos@\@ne\eql@shape@amount@z@}
2473 \def\eql@shape@layoutcenter@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
2474 \let\eql@shape@layoutcenter@default\eql@shape@layoutcenter@center
2475 \def\eql@shape@layoutleft@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2476 \def\eql@shape@layoutleft@center{\eql@shape@pos@\@ne\eql@shape@amount@z@}
2477 \def\eql@shape@layoutleft@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
2478 \let\eql@shape@layoutleft@default\eql@shape@layoutleft@left
```

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


```

2479 \def\eqL@shape@layoutcenter@first{%
2480   \eqL@shape@pos@z@
2481   \eqL@shape@amount@z@
2482   \ifnum\eqL@totalrows@>\@ne
2483     \ifnum\eqL@row@=\@ne
2484       \eqL@shape@amount@\eqL@indent@
2485     \fi
2486   \fi
2487 }
2488 \def\eqL@shape@layoutleft@first{%
2489   \eqL@shape@pos@z@
2490   \eqL@shape@amount@z@
2491   \ifnum\eqL@totalrows@>\@ne
2492     \ifnum\eqL@row@=\@ne
2493       \eqL@shape@amount@\eqL@indent@
2494     \fi
2495   \fi
2496 }

```

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:

```

2497 \def\eqL@shape@layoutcenter@hanging{%
2498   \eqL@shape@pos@z@
2499   \eqL@shape@amount@\eqL@indent@
2500   \ifnum\eqL@totalrows@>\@ne
2501     \ifnum\eqL@row@=\@ne
2502       \eqL@shape@amount@z@
2503     \fi
2504   \fi
2505 }
2506 \def\eqL@shape@layoutleft@hanging{%
2507   \eqL@shape@pos@z@
2508   \eqL@shape@amount@z@
2509   \ifnum\eqL@totalrows@>\@ne
2510     \ifnum\eqL@row@=\@ne
2511       \eqL@shape@amount@-\eqL@indent@
2512     \fi
2513   \fi
2514 }

```

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:

```

2515 \def\eqL@shape@layoutcenter@steps{%
2516   \eqL@shape@amount@z@
2517   \eqL@shape@pos@one
2518   \ifnum\eqL@totalrows@>\@ne
2519     \ifnum\eqL@row@=\@ne
2520       \eqL@shape@pos@z@
2521     \fi
2522     \ifnum\eqL@row@=\eqL@totalrows@
2523       \eqL@shape@pos@tw@
2524     \fi
2525   \fi
2526 }
2527 \def\eqL@shape@layoutleft@steps{%

```

```

2528 \eql@shape@pos@\z@
2529 \eql@shape@amount@\z@
2530 \ifnum\eql@totalrows@>\@ne
2531   \ifnum\eql@row@=\@ne
2532     \eql@shape@amount@-\eql@indent@
2533   \fi
2534   \ifnum\eql@row@=\eql@totalrows@
2535     \eql@shape@amount@\eql@indent@
2536   \fi
2537 \fi
2538 }

```

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

```

2539 \let\eql@shape@eval\undefined
2540 \def\eql@shape@select{%
2541   \expandafter\let\expandafter\eql@shape@eval
2542     \csname eql@shape%
2543       @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
2544       @\eql@shape@mode\endcsname
2545 }

```

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

`\eql@shape@alignright`

`\eql@shape@aligncenter`

```

2546 \protected\def\eql@shape@alignleft{%
2547   \global\eql@append\eql@cell@container{\eql@shape@pos@\z@}%
2548   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2549 \protected\def\eql@shape@aligncenter{%
2550   \global\eql@append\eql@cell@container{\eql@shape@pos@\@ne}%
2551   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2552 \protected\def\eql@shape@alignright{%
2553   \global\eql@append\eql@cell@container{\eql@shape@pos@\tw@}%
2554   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2555 \def\eql@shape@align@testpar#1{%
2556   \eql@ifstar@tight{#1[\eql@indent@]}%
2557   {\eql@ifnextgobble@tight{!}{#1[-\eql@indent@]}%
2558   {\eql@testopt@tight{#1}\z@}}
2559 \def\eql@shape@alignamount@opt[#1]{\eql@shape@alignamount@set{#1}}

```

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

```

2560 \protected\def\eql@shape@alignamount{%
2561   \eql@ampprotecttwo\eql@ifstar@tight
2562   \eql@shape@alignamount@set\eql@shape@alignamount@add}
2563 \def\eql@shape@alignamount@add#1{%
2564   \global\eql@appendexpand\eql@cell@container{%
2565     \advance\eql@shape@amount@\the\glueexpr#1\relax\relax}}
2566 \def\eql@shape@alignamount@set#1{%
2567   \global\eql@appendexpand\eql@cell@container{%
2568     \eql@shape@amount@\the\glueexpr#1\relax\relax}}
2569 \def\eql@shape@align@enable{%
2570   \let\shoveleft\eql@shape@alignleft
2571   \let\shovecenter\eql@shape@aligncenter
2572   \let\shoveright\eql@shape@alignright
2573   \let\shoveby\eql@shape@alignamount
2574 }

```

TODO: describe

```
2575 \protected\def\eq@shape@align@default{%
2576   \eq@warn@here{\shove...}%
2577   \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2578 \protected\def\eq@shape@align@amount@default{%
2579   \eq@warn@here{\shove...}%
2580   \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2581 \def\eq@shape@align@disable{%
2582   \let\shoveleft\eq@shape@align@default
2583   \let\shovecenter\eq@shape@align@default
2584   \let\shoveright\eq@shape@align@default
2585   \let\shoveby\eq@shape@align@amount@default
2586 }
```

9.3 Width Data

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

```
2587 \newdimen\eq@tagwidth@block@
2588 \newdimen\eq@tagheight@block@
2589 \newdimen\eq@tagdepth@block@
```

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

```
2590 \let\eq@dimensions@tab\@empty
```

`\eq@dimensions@reset`

```
2591 \def\eq@dimensions@reset{%
2592   \let\eq@dimensions@tab\@empty
2593   \eq@tagwidth@max@z@
2594   \eq@tagrows@z@
2595 }
```

`\eq@dimensions@add`

```
2596 \def\eq@dimensions@add#1{%
2597   \global\eq@appendexpand\eq@dimensions@tab{#1}%
2598 }
```

`\eq@dimensions@addreg`

```
2599 \def\eq@dimensions@addreg#1{#1\the#1\relax}
```

`\eq@dimensions@startrow`

```
2600 \def\eq@dimensions@startrow{%
2601   \eq@dimensions@add{\eq@dimensions@addreg\eq@row@}%
2602 }
```

`\eq@dimensions@savecell`

```
2603 \def\eq@dimensions@savecell{%
2604   \eq@dimensions@add{%
2605     \eq@dimensions@addreg\eq@shape@pos@
2606     \eq@dimensions@addreg\eq@cellwidth@
2607     \eq@dimensions@addreg\eq@shape@amount@
2608     \noexpand\eq@dimensions@cellcall
```

```

2609 }%
2610 }

```

l@dimensions@savesep

```

2611 \def\eqldimensions@savesep{%
2612 \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2613 }

```

ql@dimensions@endrow

```

2614 \def\eqldimensions@endrow{%
2615 \eqldimensions@add{,%
2616 \eqldimensions@addreg\eqldtagwidth@
2617 \eqldimensions@addreg\eqldline@height@
2618 \eqldimensions@addreg\eqldline@depth@
2619 \eqldimensions@addreg\eqldline@interline@
2620 ;}%
2621 }

```

ensions@saveblocktag

```

2622 \def\eqldimensions@saveblocktag{%
2623 \eqldimensions@add{\eqldrow@0\relax,%
2624 \eqldtagwidth@block@the\eqldtagwidth@\relax
2625 \eqldtagheight@block@the\ht\eqldtagbox@\relax
2626 \eqldtagdepth@block@the\dp\eqldtagbox@\relax
2627 \eqldimensions@addreg\eqldtagpos@shift@
2628 \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
2629 \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
2630 ;}%
2631 \global\eqldtagwidth@max@\eqldtagwidth@
2632 \global\eqldtagrows@\@ne
2633 }

```

sions@savenoblocktag

```

2634 \def\eqldimensions@savenoblocktag{%
2635 \eqldimensions@add{\eqldrow@0\relax,;%
2636 }

```

\eqldimensions@for

```

2637 \def\eqldimensions@for#1{%
2638 \def\eqldimensions@forall{#1}%
2639 \expandafter\eqldimensions@forstep\eqldimensions@tab
2640 }

```

l@dimensions@forstep

```

2641 \def\eqldimensions@forstep\eqldrow@#1\relax#2,#3;{%
2642 \eqldrow@#1\relax
2643 \ifnum\eqldrow@=\z@\else
2644 #3%
2645 \def\eqldimensions@cells{#2}%
2646 \eqldimensions@forall
2647 \expandafter\eqldimensions@forstep
2648 \fi
2649 }

```

`\eql@dimensions@get`

```
2650 \def\eql@dimensions@get#1{%
2651   \eql@row@#1\relax
2652   \expandafter\eql@dimensions@getdef\expandafter{\the\eql@row@}%
2653   \expandafter\eql@dimensions@getparse\eql@dimensions@tab\@nil
2654 }
```

`\eql@dimensions@getdef`

```
2655 \def\eql@dimensions@getdef#1{%
2656   \def\eql@dimensions@getparse
2657     ##1\eql@row@#1\relax##2,##3;##4\@nil{%
2658     ##3%
2659     \def\eql@dimensions@cells{##2}%
2660   }%
2661 }
```

`\eql@colwidth@tab`

```
2662 \let\eql@colwidth@tab\@empty
```

`\eql@colwidth@get`

```
2663 \def\eql@colwidth@get#1{%
2664   \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2665 }
```

`\eql@colwidth@save`

```
2666 \def\eql@colwidth@save#1{%
2667   \edef\eql@colwidth@tab{%
2668     \noexpand\or\the#1%
2669     \unexpanded\expandafter{\eql@colwidth@tab}%
2670   }%
2671 }
```

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

```
2672 \def\eql@dimensions@calc{%
2673   \eql@column@\z@
2674   \eql@line@pos@\z@
2675   \eql@line@possep@\z@
2676   \eql@line@avail@\eql@totalwidth@
2677   \eql@line@availsep@\eql@intercolumns@
2678   \eql@line@width@\z@
2679   \eql@line@widthsep@\z@
2680   \let\eql@dimensions@cellcall\eql@dimensions@calc@call
2681   \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
2682   \eql@dimensions@cells
2683 }
```

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

```

2684 \def\eqldimensions@calc@callsep{%
2685   \advance\eqlline@possep@\@ne
2686 }%

```

`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?!)

```

2687 \def\eqldimensions@calc@call{%
2688   \advance\eqlline@column@\@ne
2689   \ifnum\eqlline@totalcolumns@=\@ne
2690     \dimen@eqlline@totalwidth@
2691   \else
2692     \dimen@eqlline@colwidth@get\eqlline@column@\relax
2693   \fi
2694   \ifdim\eqlline@cellwidth@>\z@
2695     \ifdim\eqlline@width@=\z@
2696       \eqlline@avail@\eqlline@pos@
2697       \eqlline@availsep@\eqlline@possep@
2698       \ifcase\eqlline@shape@pos@
2699         \or
2700           \advance\eqlline@avail@\dimexpr
2701             (\dimen@-\eqlline@cellwidth@+\eqlline@centeroffset@)/\tw@\relax
2702         \or
2703           \advance\eqlline@avail@\dimexpr\dimen@-\eqlline@cellwidth@\relax
2704         \fi
2705         \advance\eqlline@avail@\eqlline@shape@amount@
2706       \fi
2707       \eqlline@width@\eqlline@pos@
2708       \eqlline@widthsep@\eqlline@possep@
2709       \ifcase\eqlline@shape@pos@
2710         \advance\eqlline@width@\eqlline@cellwidth@
2711       \or
2712         \advance\eqlline@width@\dimexpr
2713           (\dimen@+\eqlline@cellwidth@+\eqlline@centeroffset@)/\tw@\relax
2714       \or
2715         \advance\eqlline@width@\dimen@
2716       \fi
2717       \advance\eqlline@width@\eqlline@shape@amount@
2718     \fi
2719     \advance\eqlline@pos@\dimen@
2720 }

```

9.4 Best Line Selection

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

```

2721 \let\eqlnumbering@best@auto\eq>false

```

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

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

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

```

2722 \newcount\eqlnumbering@best@row@
2723 \newdimen\eqlnumbering@best@space@
2724 \let\eqlnumbering@best@use\eq>false

```

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

```

2725 \def\eqlnumbering@best@find{%

```

```

2726 \eql@numbering@best@row@z@
2727 \eql@numbering@best@space@z@
2728 \eql@dimensions@for{%
2729   \eql@dimensions@calc
2730   \ifdefined\eql@tagsleft
2731     \dimen@\eql@line@avail@
2732   \else
2733     \dimen@\dimexpr\eql@totalwidth@-\eql@line@width@\relax
2734   \fi
2735   \ifdim\dimen@>\eql@numbering@best@space@
2736     \eql@numbering@best@row@\eql@row@
2737     \eql@numbering@best@space@\dimen@
2738   \fi
2739 }%
2740 \ifnum\eql@numbering@best@row@>z@
2741   \eql@tagpos@row@\eql@numbering@best@row@
2742   \let\eql@tagpos@continuous\eql@false
2743   \eql@tagpos@prevrow@z@
2744 \fi
2745 }

```

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

```

2746 \def\eql@numbering@best@test#1{%
2747   \eql@dimensions@get#1%
2748   \eql@dimensions@calc
2749   \ifdefined\eql@tagsleft
2750     \dimen@\dimexpr\eql@line@avail@
2751       +\eql@marginleft@+\eql@line@availsep@\eql@colsep@\relax
2752   \else
2753     \dimen@\dimexpr\displaywidth-\eql@line@width@
2754       -\eql@marginleft@-\eql@line@widthsep@\eql@colsep@\relax
2755   \fi
2756   \ifdim\dimen@<\eql@tagwidth@block@
2757     \let\eql@numbering@best@use\eql@true
2758   \fi
2759 }

```

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

```

2760 \def\eql@numbering@best@eval{%
2761   \ifdefined\eql@numbering@best@auto
2762     \ifdefined\eql@numbering@best@use\else
2763       \ifdefined\eql@numbering@multi\else
2764         \ifnum\eql@tagpos@row@>z@
2765           \eql@numbering@best@test\eql@tagpos@row@
2766         \fi
2767         \ifnum\eql@tagpos@prevrow@>z@
2768           \eql@numbering@best@test\eql@tagpos@prevrow@
2769         \fi
2770       \fi
2771     \fi
2772   \fi
2773   \ifdefined\eql@numbering@best@use
2774     \eql@numbering@best@find
2775   \fi
2776 }

```

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

```
2777 \def\eql@adjust@calc@tagmargin{%
2778   \ifdefined\eql@tagmargin@val
2779     \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2780   \else
2781     \eql@tagmargin@\eql@tagwidth@max@
2782     \ifdim\eql@tagmargin@>\z@
2783       \advance\eql@tagmargin@-\eql@tagsepmin@
2784     \fi
2785   \fi

2786   \dimen@\eql@tagrows@p@
2787   \ifnum\eql@totalrows@=\@ne
2788     \ifnum\eql@tagrows@=\@ne
2789       \advance\dimen@1sp\relax
2790     \fi
2791   \fi
2792   \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
2793     \eql@tagmargin@\z@
2794   \fi

2795   \@tempdima\dimexpr\displaywidth
2796     -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
2797   \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
2798   \ifdim\@tempdimb>\z@
2799     \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
2800       \eql@tagmargin@\z@
2801     \fi
2802   \fi
2803 }
```

9.6 Single Column

ql@adjust@calc@lines

```
2804 \def\eql@adjust@calc@lines{%
2805   \eql@totalcolumns@\@ne
2806   \eql@intercolumns@\z@
2807   \eql@colsep@\z@
2808   \ifdefined\eql@layoutleft
2809     \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2810     \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2811     \ifdim\eql@marginleft@<\eql@marginleft@min@
2812       \eql@marginleft@\eql@marginleft@min@
2813     \fi
2814     \dimen@\glueexpr\eql@layoutleftmarginmax\relax
2815     \ifdim\eql@marginleft@>\dimen@
2816       \eql@marginleft@\dimen@
2817     \fi
2818     \eql@marginright@\z@
2819     \eql@centeroffset@\z@
2820   \else
```



```

2821 \eql@adjust@calc@tagmargin
2822 \ifdefined\eql@paddingleft@val
2823 \eql@marginleft@dimexpr
2824 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2825 -\glueexpr\eql@paddingleft@val\relax\relax
2826 \ifdim\eql@marginleft@<\z@
2827 \eql@marginleft@\z@
2828 \fi
2829 \else
2830 \eql@marginleft@\z@
2831 \fi
2832 \ifdefined\eql@paddingright@val
2833 \eql@marginright@dimexpr
2834 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2835 -\glueexpr\eql@paddingright@val\relax\relax
2836 \ifdim\eql@marginright@<\z@
2837 \eql@marginright@\z@
2838 \fi
2839 \else
2840 \eql@marginright@\z@
2841 \fi
2842 \ifdim\eql@tagmargin@>\z@
2843 \ifdefined\eql@tagsleft
2844 \ifdim\eql@marginleft@<\eql@tagsepmin@
2845 \eql@marginleft@\eql@tagsepmin@
2846 \fi
2847 \advance\eql@marginleft@\eql@tagmargin@
2848 \advance\eql@centeroffset@\eql@tagmargin@
2849 \else
2850 \ifdim\eql@marginright@<\eql@tagsepmin@
2851 \eql@marginright@\eql@tagsepmin@
2852 \fi
2853 \advance\eql@marginright@\eql@tagmargin@
2854 \advance\eql@centeroffset@-\eql@tagmargin@
2855 \fi
2856 \fi
2857 \eql@marginleft@min@\z@
2858 \eql@centeroffset@dimexpr\eql@marginright@-\eql@marginleft@
2859 \ifdefined\eql@tagsleft+\else-\fi\eql@tagmargin@\relax
2860 \fi

2861 \eql@totalwidth@dimexpr\displaywidth
2862 -\eql@marginleft@-\eql@marginright@\relax
2863 }

```

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.

`\eql@adjust@calc@columns` Main method to adjust column placement and spacing:

```
2864 \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:

```
2865 \ifnum\eq@totalcolumns@<\thr@@
2866   \eq@totalcolumns@\tw@
2867   \let\eq@columns@fulllength\eq@false
2868 \fi
```

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

```
2869 \eq@intercolumns@\numexpr(\eq@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2870 \eq@colsepmin@\glueexpr\eq@colsepmin@val\relax
```

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

```
2871 \ifdefined\eq@layoutleft
2872   \eq@marginleft@\glueexpr\eq@layoutleftmargin\relax
2873   \eq@marginleft@min@\glueexpr\eq@layoutleftmarginmin\relax
2874   \ifdim\eq@marginleft@<\eq@marginleft@min@
2875     \eq@marginleft@\eq@marginleft@min@
2876   \fi
2877 \else
```

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

```
2878   \eq@adjust@calc@tagmargin
2879   \ifdefined\eq@columns@fulllength
2880     \ifdim\eq@tagmargin@>\z@
2881       \advance\eq@tagmargin@\eq@tagsepmin@
2882     \fi
2883   \fi
2884   \ifdim\eq@tagmargin@>\dimexpr\displaywidth-\eq@totalwidth@
2885     -\eq@intercolumns@\eq@colsepmin@\relax
2886     \eq@tagmargin@\z@
2887   \fi
2888   \eq@marginleft@min@\z@
2889 \fi
```

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

```
2890 \ifnum\eq@intercolumns@>\z@
```

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

```
2891   \eq@colsep@\dimexpr\displaywidth-\eq@totalwidth@\relax
2892   \ifdefined\eq@layoutleft
2893     \advance\eq@colsep@-\eq@marginleft@
2894   \else
2895     \advance\eq@colsep@-\eq@tagmargin@
2896   \fi
2897   \count@\eq@intercolumns@
2898   \ifdefined\eq@columns@fulllength\else
2899     \ifdefined\eq@layoutleft
2900       \advance\count@\@ne
2901     \else
2902       \advance\count@\tw@
2903   \fi
```

```

2904 \fi
2905 \divide\eql@colsep@\count@

```

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

```

2906 \ifdim\eql@colsep@<\eql@colsepmin@
2907 \eql@colsep@\eql@colsepmin@
2908 \else
2909 \ifdefined\eql@columns@fulllength\else
2910 \dimen@\glueexpr\eql@colsepmax@val\relax
2911 \ifdim\eql@colsep@>\dimen@
2912 \eql@colsep@\dimen@
2913 \fi
2914 \fi
2915 \fi
2916 \else

```

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

```

2917 \eql@colsep@\eql@colsepmin@
2918 \fi

```

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

```

2919 \ifdefined\eql@layoutleft

```

Set the default value:

```

2920 \ifdim\eql@colsep@=\eql@colsepmin@

```

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

```

2921 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
2922 -\eql@intercolumns@\eql@colsep@\relax
2923 \ifdim\dimen@<\eql@marginleft@
2924 \ifdim\dimen@<\eql@marginleft@min@
2925 \eql@marginleft@\eql@marginleft@min@
2926 \else
2927 \eql@marginleft@\dimen@
2928 \fi
2929 \fi
2930 \fi
2931 \else

```

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

```

2932 \ifdefined\eql@columns@fulllength
2933 \eql@marginleft@\z@

```

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

```

2934 \else
2935 \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@
2936 -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
2937 \ifdim\eql@marginleft@<\z@
2938 \eql@marginleft@\z@
2939 \fi
2940 \fi

```

Add tag margin in case of left tags:

```
2941 \ifdefined\eq@tagsleft
2942 \advance\eq@marginleft\eq@tagmargin@
2943 \fi
2944 \fi
```

Find the best row for tag placement:

```
2945 \eq@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:

```
2946 \ifdefined\eq@tagsleft
2947 \let\eq@adjust@columns@test\eq@adjust@columns@test@tagsleft
2948 \else
2949 \let\eq@adjust@columns@test\eq@adjust@columns@test@tagsright
2950 \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

```
2951 \ifdefined\eq@numbering@multi
2952 \eq@dimensions@for{%
2953 \ifdim\eq@tagwidth@>\z@
2954 \eq@dimensions@calc
2955 \eq@adjust@columns@test
2956 \fi
2957 }%
2958 \else
2959 \ifnum\eq@tagpos@row@>\z@
2960 \ifnum\eq@tagpos@row@>\eq@totalrows@\else
2961 \eq@dimensions@get\eq@tagpos@row@
2962 \eq@tagwidth@\eq@tagwidth@block@
2963 \eq@dimensions@calc
2964 \eq@adjust@columns@test
2965 \fi
2966 \fi
2967 \ifnum\eq@tagpos@prevrow@>\z@
2968 \eq@dimensions@get\eq@tagpos@prevrow@
2969 \eq@tagwidth@\eq@tagwidth@block@
2970 \eq@dimensions@calc
2971 \eq@adjust@columns@test
2972 \fi
2973 \fi
```

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

```
2974 \advance\eq@totalwidth@\dimexpr
2975 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
2976 }
```

Placement for Right Tags.

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

```
2977 \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?

```
2978 \@tempdima\dimexpr\displaywidth-\eql@line@width@-\eql@tagwidth@\relax
```

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

```
2979 \ifdim\@tempdima<\dimexpr
2980 \eql@marginleft@+\eql@line@widthsep@\eql@colsep@\relax
```

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

```
2981 \ifdim\@tempdima<\dimexpr
2982 \eql@marginleft@min@+\eql@line@widthsep@\eql@colsepmin@\relax\else
```

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

```
2983 \eql@adjust@columns@modify@tagsright
2984 \fi
2985 \fi
2986 }
```

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

```
2987 \def\eql@adjust@columns@modify@tagsright{%
```

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

```
2988 \ifnum\eql@line@widthsep@>\z@
2989 \dimen@\@tempdima
2990 \count@\eql@line@widthsep@
2991 \ifdefined\eql@layoutleft
2992 \advance\dimen@-\eql@marginleft@
2993 \else
2994 \ifdefined\eql@columns@fulllength\else
2995 \advance\count@\@ne
2996 \fi
2997 \fi
2998 \divide\dimen@\count@
```

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

```
2999 \ifdim\dimen@<\eql@colsep@
3000 \ifdim\dimen@<\eql@colsepmin@
3001 \eql@colsep@\eql@colsepmin@
3002 \else
3003 \eql@colsep@\dimen@
3004 \fi
3005 \fi
3006 \fi
```

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

```
3007 \dimen@\dimexpr\@tempdima-\eql@line@widthsep@\eql@colsep@\relax
3008 \ifdim\eql@marginleft@>\dimen@
```

```

3009   \eql@marginleft@dimen@
3010   \fi
3011 }

```

Placement for Left Tags.

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

```
3012 \def\eql@adjust@columns@test@tagsleft{%
```

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

```

3013   \count@numexpr\eql@intercolumns@-\eql@line@availsep@relax
3014   \@tempdima\dimexpr\eql@tagwidth@-\eql@line@avail@relax

```

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

```

3015   \ifdim\@tempdima>\dimexpr
3016     \eql@marginleft@+\eql@line@availsep@eql@colsep@relax

```

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

```

3017   \ifdim\eql@tagwidth@<%
3018     \ifdefined\eql@layoutleft
3019       \glueexpr\eql@layoutleftmarginmax@relax
3020     \else
3021       \displaywidth
3022     \fi

```

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

```

3023     \ifdim\@tempdima>\dimexpr
3024       \displaywidth-\eql@totalwidth@-\count@eql@colsepmin@relax\else

```

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

```

3025     \eql@adjust@columns@modify@tagsleft
3026     \fi
3027   \fi
3028 \fi
3029 }

```

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

```
3030 \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.

```

3031   \ifnum\count@>\z@
3032     \dimen@dimexpr\displaywidth-\eql@totalwidth@-\@tempdima@relax
3033     \ifdefined\eql@columns@fulllength@else
3034       \advance\count@ \@ne
3035     \fi
3036     \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.

```

3037 \ifdim\dimen@<\eql@colsep@
3038 \ifdim\dimen@<\eql@colsepmin@
3039 \dimen@\eql@colsepmin@
3040 \fi
3041 \advance\dimen@-\eql@colsep@
3042 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3043 \advance\eql@colsep@\dimen@
3044 \fi
3045 \fi

```

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

```

3046 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3047 \ifdim\eql@marginleft@<\dimen@
3048 \eql@marginleft@\dimen@
3049 \fi
3050 }

```

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:

```

3051 \ifdefined\inf@bad\else%
3052 \newcount\inf@bad
3053 \inf@bad1000000\relax
3054 \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`:

```

3055 \let\eql@restore@hfuzz\@empty
3056 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}

```

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

```

3057 \newcount\eql@alignbadness@
3058 \newcount\eql@tagbadness@
3059 \newcount\eql@arrange@badness@
3060 \eql@alignbadness@\inf@bad
3061 \eql@tagbadness@\inf@bad

```

10.2 Arrangement Methods

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

```

3062 \def\eq@arrange@try#1{%
3063   \ifdim#1>\dimexpr\displaywidth-\eq@cellwidth@relax
3064     \setbox\eq@cellbox@hbox to\displaywidth{%
3065       \unhbox\eq@cellbox@unkern\kern#1}%
3066     \eq@arrange@badness@badness
3067   \else
3068     \eq@arrange@badness@m@ne
3069   \fi
3070 }

```

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

```

3071 \def\eq@arrange@print#1#2{%
3072   \eq@restore@hfuzz
3073   \if@eqnsw
3074     \ifdefined\eq@tagsleft
3075       \eq@tagbox@print@tagsleft
3076     \fi
3077   \fi
3078   \hbox to\displaywidth{%
3079     #1%
3080     \unhbox\eq@cellbox@unkern
3081     #2%
3082     \eq@tagging@mathaddlast
3083   }%
3084   \if@eqnsw
3085     \ifdefined\eq@tagsleft\else
3086       \eq@tagbox@print@tagsright
3087     \fi
3088   \fi
3089 }

```

`\eq@arrange@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 `\eq@display@firstavail@` to the minimum left available space we can guarantee:

```

3090 \def\eq@arrange@print@alignleft#1#2{%
3091   \eq@display@firstavail@set{\dimexpr#1relax}%
3092   \eq@arrange@print{\kern#1}{\kern#2}%
3093 }

3094 \def\eq@arrange@print@alignright#1#2{%
3095   \eq@display@firstavail@set{\dimexpr\displaywidth-\eq@cellwidth@-#2relax}%
3096   \eq@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3097 }

3098 \def\eq@arrange@print@aligncenter#1{%
3099   \eq@display@firstavail@set{\dimexpr
3100     (\displaywidth-\eq@cellwidth@+#1)/\tw@relax}%
3101   \ifdim#1>\z@
3102     \eq@arrange@print{\kern#1\hfil}{}%
3103   \else
3104     \eq@arrange@print{\hfil}{\kern-#1}%
3105   \fi
3106 }

```

`\eq@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:

```
3107 \def\eql@arrange@init{%
3108   \eql@save@hfuzz
3109   \hfuzz\maxdimen
3110   \eql@shape@select
3111 }
```

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

```
3112 \def\eql@arrange@print@line{%
3113   \eql@tagging@tagaddbox
3114   \csname eql@arrange%
3115     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3116     @init\endcsname
3117   \csname eql@arrange%
3118     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3119     @\ifdefined\eql@tagpos@reserve
3120       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3121       notag\fi\endcsname
3122 }
```

10.3 Central Alignment

TODO: describe

```
3123 \def\eql@arrange@aligncenter@init{%
3124   \eql@tagging@aligncenter
3125   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3126     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3127 }
```

TODO: describe

```
3128 \def\eql@arrange@aligncenter@notag{%
3129   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3130     \ifdim\eql@line@offset@<\eql@marginleft@min@
3131       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3132     \else
3133       \eql@line@offset@
3134     \fi
3135   \eql@arrange@print@aligncenter\eql@line@offset@
3136 \else
3137   \ifdim\eql@line@offset@<\eql@marginleft@min@
3138     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3139   \else
3140     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3141   \fi
3142 \fi
3143 }
```

TODO: describe

```
3144 \def\eql@arrange@aligncenter@tagsright{%
3145   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3146     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3147       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
```

```

3148     \else
3149         \dimexpr\tw\@eq\tagwidth@+\@eq\line@offset@\relax
3150     \fi
3151     \@eq\arrange@print@aligncenter\@eq\line@offset@
3152 \else
3153     \@eq\arrange@try{\dimexpr\@eq\tagwidth@+\@eq\marginleft@min@\relax}%
3154     \ifnum\@eq\arrange@badness@<\@eq\tagbadness@
3155         \ifdim\@eq\line@offset@<\dimexpr\@eq\marginleft@min@-\@eq\tagwidth@\relax
3156             \@eq\arrange@print@alignleft\@eq\marginleft@min@\@eq\tagwidth@
3157         \else
3158             \@eq\arrange@print@alignright\@eq\marginleft@min@\@eq\tagwidth@
3159         \fi
3160     \else
3161         \let\@eq\tagpos@reserve\@eq>false
3162         \@eq\arrange@aligncenter@notag
3163     \fi
3164 \fi
3165 }

3166 \def\@eq\arrange@aligncenter@tagsleft{%
3167     \ifdim\@eq\tagwidth@>\@eq\marginleft@min@
3168         \ifdim\dimexpr\displaywidth-\@eq\cellwidth@\relax>%
3169             \ifdim\@eq\line@offset@<\@eq\tagwidth@
3170                 \dimexpr\tw\@eq\tagwidth@-\@eq\line@offset@\relax
3171             \else
3172                 \@eq\line@offset@
3173             \fi
3174             \@eq\arrange@print@aligncenter\@eq\line@offset@
3175         \else
3176             \@eq\arrange@try\@eq\tagwidth@
3177             \ifnum\@eq\arrange@badness@<\@eq\tagbadness@
3178                 \ifdim\@eq\line@offset@<\@eq\tagwidth@
3179                     \@eq\arrange@print@alignleft\@eq\tagwidth@\z@
3180                 \else
3181                     \@eq\arrange@print@alignright\@eq\tagwidth@\z@
3182                 \fi
3183             \else
3184                 \let\@eq\tagpos@reserve\@eq>false
3185                 \@eq\arrange@aligncenter@notag
3186             \fi
3187         \fi
3188     \else
3189         \@eq\arrange@aligncenter@notag
3190     \fi
3191 }

```

10.4 Left Alignment

```

3192 \def\@eq\arrange@alignleft@init{%
3193     \@eq\tagging@alignleft
3194     \@eq\line@offset@\dimexpr\@eq\marginleft@+\@eq\shape@amount@\relax
3195     \ifdim\@eq\line@offset@<\@eq\marginleft@min@
3196         \@eq\line@offset@\@eq\marginleft@min@
3197     \fi
3198 }

3199 \def\@eq\arrange@alignleft@notag{%
3200     \ifdim\@eq\line@offset@>\@eq\marginleft@min@
3201         \@eq\arrange@try\@eq\line@offset@

```

```

3202 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3203 \eqL@arrange@print@alignleft\eqL@line@offset@z@
3204 \else
3205 \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3206 \fi
3207 \else
3208 \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3209 \fi
3210 }

3211 \def\eqL@arrange@alignleft@tagsright{%
3212 \eqL@arrange@try{\dimexpr\eqL@line@offset@+\eqL@tagwidth@relax}%
3213 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3214 \eqL@arrange@print@alignleft\eqL@line@offset@eqL@tagwidth@
3215 \else
3216 \ifdim\eqL@line@offset@>\eqL@marginleft@min@
3217 \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3218 \fi
3219 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3220 \eqL@arrange@print@alignright\eqL@marginleft@min@eqL@tagwidth@
3221 \else
3222 \let\eqL@tagpos@reserve\eqL@false
3223 \eqL@arrange@alignleft@notag
3224 \fi
3225 \fi
3226 }

3227 \def\eqL@arrange@alignleft@tagsleft{%
3228 \ifdim\eqL@tagwidth@>\eqL@marginleft@min@
3229 \ifdim\eqL@line@offset@>\eqL@tagwidth@
3230 \eqL@arrange@try\eqL@line@offset@
3231 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3232 \eqL@arrange@print@alignleft\eqL@line@offset@z@
3233 \else
3234 \eqL@arrange@try\eqL@tagwidth@
3235 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3236 \eqL@arrange@print@alignright\eqL@tagwidth@z@
3237 \else
3238 \let\eqL@tagpos@reserve\eqL@false
3239 \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3240 \fi
3241 \fi
3242 \else
3243 \eqL@arrange@try\eqL@tagwidth@
3244 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3245 \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3246 \else
3247 \let\eqL@tagpos@reserve\eqL@false
3248 \eqL@arrange@alignleft@notag
3249 \fi
3250 \fi
3251 \else
3252 \eqL@arrange@alignleft@notag
3253 \fi
3254 }

```

10.5 Right Alignment

```

3255 \def\eqL@arrange@alignright@init{%
3256 \eqL@tagging@alignright

```

```

3257 \eql@line@offset@\dimexpr\eql@marginright@-\eql@shape@amount@\relax
3258 \ifdim\eql@line@offset@<\z@
3259   \eql@line@offset@\z@
3260 \fi
3261 }

```

TODO: describe

```

3262 \def\eql@arrange@alignright@notag{%
3263   \ifdim\eql@line@offset@>\z@
3264     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3265     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3266       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3267     \else
3268       \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3269     \fi
3270 \else
3271   \eql@arrange@print@alignright\eql@marginleft@min@\z@
3272 \fi
3273 }

```

TODO: describe

```

3274 \def\eql@arrange@alignright@tagsright{%
3275   \ifdim\eql@line@offset@>\eql@tagwidth@
3276     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3277     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3278       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3279     \else
3280       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3281       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3282         \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3283       \else
3284         \let\eql@tagpos@reserve\eql@false
3285         \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3286       \fi
3287     \fi
3288 \else
3289   \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3290   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3291     \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3292   \else
3293     \let\eql@tagpos@reserve\eql@false
3294     \eql@arrange@alignright@notag
3295   \fi
3296 \fi
3297 }

```

TODO: describe

```

3298 \def\eql@arrange@alignright@tagsleft{%
3299   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3300     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3301     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3302       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3303     \else
3304       \ifdim\eql@line@offset@>\z@
3305         \eql@arrange@try\eql@tagwidth@
3306       \fi
3307       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3308         \eql@arrange@print@alignleft\eql@tagwidth@\z@

```

```

3309     \else
3310         \let\eql@tagpos@reserve\eql@false
3311         \eql@arrange@alignright@notag
3312     \fi
3313 \fi
3314 \else
3315     \eql@arrange@alignright@notag
3316 \fi
3317 }

```

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`

```

3318 \protected\def\eql@box@cr{\eql@srbgroup%
3319 \eql@ifnextgobble@tight~%
3320 {\let\eql@punct@line\@empty\eql@box@cr@opt}%
3321 \eql@box@cr@opt}
3322 \def\eql@box@cr@opt{\eql@ifnextchar@tight [\eql@box@cr@skip\eql@box@cr@plain]}
3323 \def\eql@box@cr@plain{\eql@sregroup\eql@box@cr@}
3324 \def\eql@box@cr@skip[#1]{\eql@sregroup
3325 \eql@box@cr@
3326 \noalign{%
3327 \vskip\glueexpr#1\relax
3328 }%
3329 }
3330 \def\eql@box@cr@f{%
3331 \eql@punct@apply@line
3332 \eql@hook@lineout
3333 \eql@box@lastcell
3334 \cr
3335 }

```

11.2 Stacked Mode

```

3336 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3337 \def\eql@box@open@stacked{%
3338 \eql@shape@align@enable
3339 \let\eql@box@lastcell\eql@box@lastcell@stacked
3340 \everycr{\noalign{%
3341 (dev)\eql@dev{starting line \the\eql@row@}%
3342 \global\advance\eql@row@\@ne
3343 }%
3344 \tabskip\z@skip
3345 \halign\bgroup
3346 &%

```

```

3347 \global\let\eqL@cell@container\@empty
3348 \setbox\eqL@cellbox@\hbox{%
3349 \eqL@strut@cell
3350 \@lign
3351 $\m@th\displaystyle
3352 \eqL@hook@colin
3353 ##%
3354 \eqL@punct@apply@col
3355 \eqL@hook@colout
3356 \eqL@tagging@mathsave
3357 $%
3358 \eqL@tagging@mathaddlast
3359 }%
3360 \ifdefined\eqL@shape@lastrow
3361 \eqL@totalrows@\eqL@row@
3362 \fi
3363 \eqL@shape@eval
3364 \eqL@cell@container
3365 \ifdefined\eqL@frame@cmd
3366 \ifcase\eqL@shape@pos@
3367 \eqL@frame@measure
3368 \advance\eqL@shape@amount@-\eqL@frame@margin@
3369 \or\or
3370 \eqL@frame@measure
3371 \advance\eqL@shape@amount@\eqL@frame@margin@
3372 \fi
3373 \eqL@frame@print
3374 \fi
3375 \ifcase\eqL@shape@pos@
3376 \kern\eqL@shape@amount@
3377 \box\eqL@cellbox@
3378 \hskip\glueexpr\eqL@paddingleft@+\eqL@paddingright@
3379 -\eqL@shape@amount@\@flushglue\relax
3380 \eqL@tagging@alignleft
3381 \or
3382 \hskip\glueexpr\eqL@paddingleft@+\eqL@shape@amount@\@flushglue\relax
3383 \box\eqL@cellbox@
3384 \hskip\glueexpr\eqL@paddingright@-\eqL@shape@amount@\@flushglue\relax
3385 \eqL@tagging@aligncenter
3386 \or
3387 \hskip\glueexpr\eqL@paddingleft@+\eqL@paddingright@
3388 +\eqL@shape@amount@\@flushglue\relax
3389 \box\eqL@cellbox@
3390 \kern-\eqL@shape@amount@
3391 \eqL@tagging@alignright
3392 \fi
3393 \tabskip\eqL@colsep@\relax
3394 \crrc
3395 \noalign{%
3396 \global\let\eqL@shape@lastrow\eqL@false
3397 \eqL@hook@blockbefore
3398 }%
3399 \eqL@hook@blockin
3400 }
3401 \def\eqL@mode@stacked{\let\eqL@box@open\eqL@box@open@stacked}

```

11.3 Aligned Mode

```

3402 \def\eq@box@lastcell@odd{%
3403   &\omit
3404   \eq@prevwidth@\wd\eq@cellbox@
3405   \let\eq@frame@cmd\eq@frame@prevcmd
3406   \ifdefined\eq@frame@cmd
3407     \eq@frame@measure
3408     \advance\eq@prevwidth@\eq@frame@margin@
3409     \eq@frame@print
3410   \fi
3411   \kern-\eq@prevwidth@
3412   \unhbox\eq@cellbox@
3413   \hfil
3414   &\omit\kern-\eq@colsep@
3415 }%
3416 \def\eq@box@lastcell@even{&\omit\kern-\eq@colsep@}

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

```

```

3460     \eql@hook@innerright
3461     \eql@class@innerright@sel
3462     ##%
3463     \eql@punct@apply@col
3464     \eql@hook@colout
3465     \eql@tagging@mathsave
3466     $%
3467     \eql@tagging@mathaddlast
3468     }%
3469     \eql@cell@container
3470     \ifdefined\eql@frame@cmd
3471     \eql@frame@measure
3472     \advance\eql@prevwidth@\eql@frame@margin@
3473     \eql@frame@print
3474     \fi
3475     \kern-\eql@prevwidth@
3476     \unhbox\eql@cellbox@
3477     \hfil
3478     \tabskip\eql@colsep@\relax
3479     \crrc
3480     \noalign{%
3481     \eql@hook@blockbefore
3482     }%
3483     \eql@hook@blockin
3484 }
3485 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

11.4 Main

```

3486 \let\eql@box@box\vcenter
3487 \let\eql@box@open@\undefined
3488 \let\eql@box@frame@\firstofone
3489 \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!

```

3490 \def\eql@box@close{%
3491     \ifvmode\else
3492     \global\let\eql@shape@lastrow\eql@true
3493     \eql@punct@apply@block
3494     \eql@box@cr@
3495     \fi
3496     \noalign{%
3497     \eql@hook@blockafter
3498     \global\let\eql@shape@lastrow\eql@false
3499     }%
3500     \eql@tagging@tablesaveinner
3501 \egroup
3502 }

```

`\eql@box@vcenter`

```

3503 \def\eql@box@vcenter#1{%
3504     \ifmmode
3505     \vcenter{#1}%

```



```

3506 \else
3507   $\m@th\vcenter{#1}$%
3508 \fi
3509 }

```

`\eql@box@start`

```

3510 \let\eql@box@endmath\eql@false
3511 \def\eql@box@start{%
3512   \relax
3513   \ifmmode
3514     \let\eql@box@endmath\eql@false
3515   \else
3516     \let\eql@box@endmath\eql@true
3517     \expandafter$%$
3518   \fi
3519   \eql@box@processopt
3520   \eql@stack@save@box
3521   \let\eql@frame@cmd\@undefined
3522   \let\eql@layoutleft\eql@false
3523   \eql@row@z@
3524   \eql@totalrows@\@M
3525   \eql@shape@select
3526   \setbox\z@\ifx\eql@box@box\vcenter
3527     \expandafter\ vbox
3528   \else
3529     \expandafter\eql@box@box
3530   \fi\bgroup
3531   \eql@display@nest
3532   \let\\eql@box@cr
3533   \eql@spread@set
3534   \eql@strut@make
3535   \eql@box@open
3536 }

```

`\eql@box@end`

```

3537 \def\eql@box@end{%
3538   \eql@box@close
3539   \egroup
3540   \eql@box@frame{%
3541     \ifdefined\eql@display@marginleft
3542       \hskip\glueexpr\eql@display@marginleft\relax
3543     \fi
3544     \ifx\eql@box@box\vcenter
3545       \eql@box@vcenter{\unvbox\z@}%
3546     \else
3547       \box\z@
3548     \fi
3549     \eql@tagging@tableadinner
3550     \ifdefined\eql@display@marginright
3551       \hskip\glueexpr\eql@display@marginright\relax
3552     \fi
3553   }%
3554   \eql@stack@restore
3555   \ifdefined\eql@box@endmath
3556     \expandafter$%$
3557   \fi
3558 }

```

11.5 Environment

`equationsbox` (*env.*)

```
3559 \newenvironment{equationsbox}{%
3560 (dev)\eql@dev@enterenv
3561 \eql@ampprotect\eql@box@testall\eql@box@start
3562 }{%
3563 \eql@box@end
3564 (dev)\eql@dev@leaveenv
3565 }

3566 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3567 \def\eql@box@parseopt{%
3568 \ifx\eql@parseopt@token[%]
3569 \let\eql@parseopt@next\eql@parseopt@opt
3570 \fi
3571 \ifx\eql@parseopt@token=%
3572 \let\eql@parseopt@next\eql@parseopt@lines
3573 \fi
3574 \ifx\eql@parseopt@token|
3575 \let\eql@parseopt@next\eql@parseopt@columns
3576 \fi
3577 \ifx\eql@parseopt@token'
3578 \let\eql@parseopt@next\eql@parseopt@punctall
3579 \fi
3580 }
```

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

```
3581 \def\eql@box@processopt{%
3582 \let\eql@box@frame\@firstofone
3583 \let\eql@display@marginleft\@undefined
3584 \let\eql@display@marginright\@undefined
3585 \eql@nextopt@process{equationsbox}%
3586 \let\eql@punct@block\eql@punct@main
3587 \let\eql@punct@main\relax
3588 \eql@colsep@\glueexpr\eql@box@colsep\relax
3589 \ifdefined\eql@paddingleft@val
3590 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3591 \else
3592 \eql@paddingleft@\z@
3593 \fi
3594 \ifdefined\eql@paddingright@val
3595 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3596 \else
3597 \eql@paddingright@\z@
3598 \fi
3599 \eql@indent@\glueexpr\eql@indent@val\relax
3600 }
```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

3601 \def\eq@single@start@native{%
3602   \eq@display@init
3603   \eq@display@print
3604   \let\raisetag\eq@raisetag@default
3605   \eq@shape@align@disable
3606   \eq@hook@eqin
3607 %   \mathopen{}%
3608 }%

```

TODO: describe

```

3609 \def\eq@single@end@native{%
3610 %   \mathclose{}%
3611   \eq@tags@container
3612   \eq@numbering@single@eval
3613   \if@eqnsw
3614     \ifdefined\eq@tagsleft
3615       \leqno
3616     \else
3617       \eqno
3618     \fi
3619   \eq@composetag@print
3620 \fi
3621 \eq@interline@container
3622 \advance\eq@belowspace@\eq@vspaceskip@
3623 \eq@display@container
3624 \eq@display@penalty
3625 \eq@display@vspace@native
3626 }%

```

12.2 Print

```

3627 \def\eq@single@start@print{%
3628   \eq@display@init
3629   \eq@display@print
3630   \eq@shape@align@enable
3631   \eq@totalrows@\@ne
3632   \eq@row@\@ne
3633   \eq@arrange@init
3634   \global\let\eq@cell@container\@empty
3635   \prevgraf\numexpr\prevgraf+\@ne\relax
3636   \setbox\eq@cellbox@\hbox\bgroup
3637     \eq@restore@hfuzz
3638     \eq@strut@cell
3639     $\m@th\displaystyle%$
3640     \eq@hook@eqin
3641 }
3642 \def\eq@single@end@print{%
3643   \eq@tagging@mathsave
3644   $%$
3645   \hfil
3646   \kern\z@
3647   \egroup
3648   \prevgraf\numexpr\prevgraf-\@ne\relax
3649   \eq@shape@eval
3650   \eq@cell@container
3651   \ifdefined\eq@frame@cmd
3652     \eq@frame@adjust

```

```

3653 \fi
3654 \eql@cellwidth@\wd\eql@cellbox@
3655 \eql@line@height@\ht\eql@cellbox@
3656 \eql@line@depth@\dp\eql@cellbox@
3657 \eql@totalwidth@\eql@cellwidth@
3658 \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
3659 \eql@topheight@\eql@line@height@
3660 \eql@bottomdepth@\eql@line@depth@

3661 \eql@tags@container
3662 \eql@numbering@single@eval
3663 \if@eqnsw
3664 \eql@tagbox@make\eql@composetag@print
3665 \eql@tagrows@\@ne
3666 \ifdefined\eql@tagpos@reserve\else
3667 \eql@tagwidth@\z@
3668 \fi
3669 \eql@tagheight@block@\ht\eql@tagbox@
3670 \eql@tagdepth@block@\dp\eql@tagbox@
3671 \else
3672 \eql@numbering@warnunused
3673 \eql@tagwidth@\z@
3674 \eql@tagrows@\z@
3675 \fi
3676 \eql@tagwidth@max@\eql@tagwidth@
3677 \eql@tagpos@single@eval
3678 \eql@tagpos@print@line@eval

3679 \eql@intercolumns@\z@
3680 \eql@adjust@calc@lines

3681 \eql@display@halign@init{ }%
3682 \halign{##\crr
3683 \noalign{\eql@display@halign@start}%
3684 \eql@arrange@print@line
3685 \cr
3686 \noalign{\eql@display@halign@end}%
3687 \eql@tagging@tablesavelines
3688 }%
3689 \eql@tagpos@print@line@end
3690 \eql@display@close
3691 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```

3692 \def\eql@lines@measure@line@begin{%
3693 (dev)\eql@dev{starting line \the\eql@row}%
3694 \eql@numbering@measure@line@begin
3695 \eql@hook@linein
3696 }

```

TODO: describe

```

3697 \def\eq@lines@measure@line@end{%
3698   \eq@punct@apply@line
3699   \eq@hook@lineout
3700 }

```

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

```

3701 \def\eq@lines@measure@cell{%
3702   \ifdefined\eq@frame@cmd
3703     \ifcase\eq@shape@pos@
3704       \eq@frame@measure
3705       \advance\eq@shape@amount@-\eq@frame@margin@
3706     \or\or
3707       \eq@frame@measure
3708       \advance\eq@shape@amount@+\eq@frame@margin@
3709     \fi
3710     \eq@frame@print
3711   \fi
3712   \eq@cellwidth@\wd\eq@cellbox@
3713   \eq@line@height@\ht\eq@cellbox@
3714   \eq@line@depth@\dp\eq@cellbox@
3715   \eq@dimensions@startrow
3716   \eq@dimensions@savecell
3717   \kern\eq@cellwidth@
3718 }

```

`\eq@lines@measure`

```

3719 \def\eq@lines@measure{%
3720 (dev)\eq@dev@enter\eq@lines@measure
3721   \eq@measure@init\eq@lines@measure@line@begin\eq@lines@measure@line@end
3722   \eq@totalrows@\@M
3723   \eq@shape@select

3724   \setbox\z@\vbox{\measuring@true\halign{%
3725     \global\let\eq@cell@container\@empty
3726     \setbox\eq@cellbox@\hbox{%
3727       \eq@strut@cell
3728       \@lign
3729       $\m@th\displaystyle
3730       \eq@hook@colin
3731       ##%
3732       \eq@punct@apply@col
3733       \eq@hook@colout
3734       $%
3735     }%
3736     \ifdefined\eq@shape@lastrow
3737       \eq@totalrows@\eq@row@
3738     \fi
3739     \eq@shape@eval
3740     \eq@cell@container
3741     \eq@lines@measure@cell
3742     \eq@measure@tag
3743     \eq@measure@endrow
3744   \crr

3745   \noalign{%
3746     \global\let\eq@shape@lastrow\eq@false
3747     \eq@hook@blockbefore

```

```

3748 }%
3749 \eql@hook@blockin
3750 \eql@scan@body
3751 \ifvmode\else
3752   \global\let\eql@shape@lastrow\eql@true
3753   \eql@punct@apply@block
3754   \eql@hook@blockout
3755   \eql@display@endline
3756   \cr
3757   \fi
3758   \omit
3759   \cr
3760   \noalign{%
3761     \eql@hook@blockafter
3762     \global\let\eql@shape@lastrow\eql@false
3763   }%
3764 }}%

3765 \eql@measure@close

3766 \setbox\z@\vbox{%
3767   \unvbox\z@
3768   \unpenalty
3769   \global\setbox\@ne\lastbox
3770 }%
3771 \eql@totalwidth@\wd\@ne

3772 (dev)\eql@dev@leave\eql@lines@measure
3773 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

3774 \def\eql@lines@adjust{%
3775   \eql@tagpos@adjust@eval
3776   \eql@adjust@calc@lines
3777   \eql@numbering@best@eval
3778 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```

3779 \def\eql@lines@print@line@begin{%
3780 (dev)\eql@dev{starting line \the\eql@row}%
3781   \eql@numbering@print@line@begin
3782   \eql@hook@linein
3783 }

```

TODO: describe

```

3784 \def\eql@lines@print@line@end{%
3785   \eql@punct@apply@line
3786   \eql@hook@lineout
3787 }

```

TODO: describe

```
3788 \def\eq@lines@print@line@adjust{%
3789   \ifdefined\eq@frame@cmd
3790     \ifcase\eq@shape@pos@
3791       \eq@frame@measure
3792       \advance\eq@shape@amount@-\eq@frame@margin@
3793     \or\or
3794       \eq@frame@measure
3795       \advance\eq@shape@amount@+\eq@frame@margin@
3796     \fi
3797     \eq@frame@adjust
3798   \fi
3799   \eq@cellwidth@\wd\eq@cellbox@
3800   \eq@line@height@\ht\eq@cellbox@
3801   \eq@line@depth@\dp\eq@cellbox@
3802   \eq@numbering@print@line@eval
3803   \if@eqnsw
3804     \eq@tagbox@make\eq@composetag@print
3805   \fi
3806   \eq@tagpos@print@line@eval
3807   \eq@arrange@print@line
3808   \eq@tagpos@print@line@end
3809 }
```

TODO: describe

```
3810 \def\eq@lines@print{%
3811 (dev)\eq@dev@enter\eq@lines@print
3812   \eq@arrange@init
3813   \eq@display@halign@init\eq@lines@print@line@begin
3814   \eq@display@halign@letcr\eq@lines@print@line@end
3815   \tabskip\z@skip

3816   \halign{%
3817     \global\let\eq@cell@container\@empty
3818     \setbox\eq@cellbox@\hbox{%
3819       \eq@restore@hfuzz
3820       \eq@strut@cell
3821       \@lign
3822       $\m@th\displaystyle
3823       \eq@hook@colin
3824       ##%
3825       \eq@punct@apply@col
3826       \eq@hook@colout
3827       \eq@tagging@mathsave
3828       $%
3829       \hfil
3830       \kern\z@
3831     }%
3832     \eq@shape@eval
3833     \eq@cell@container
3834     \eq@lines@print@line@adjust
3835   \crr

3836   \noalign{%
3837     \eq@display@halign@start
3838     \eq@numbering@print@block@begin
3839     \eq@hook@blockbefore
3840   }%
```

```

3841 \eql@hook@blockin
3842 \eql@scan@body
3843 \ifvmode\else
3844 \relax
3845 \eql@punct@apply@block
3846 \eql@hook@blockout
3847 \eql@display@endline
3848 \cr
3849 \fi
3850 \noalign{%
3851 \eql@hook@blockafter
3852 \eql@display@halign@end
3853 (dev)\eql@dev@leave\eql@lines@print
3854 }%
3855 \eql@tagging@tablesavelines
3856 }%
3857 }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```

\eql@columns@add@amp
@columns@completerow
3858 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3859 \def\eql@columns@completerow{%
3860 \count@numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
3861 \edef\eql@tmp{%
3862 \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3863 \eql@tmp
3864 }

3865 \def\eql@columns@overfull{%
3866 \dimen@eql@line@width@
3867 \advance\dimen@-\hfuzz
3868 \ifdim\dimen@>\displaywidth
3869 \setbox\z@\hbox to\displaywidth{\hbox to\eql@line@width@{\hfil}}%
3870 \wd\z@\z@
3871 \ht\z@\eql@line@height@
3872 \dp\z@\eql@line@depth@
3873 \box\z@
3874 \fi
3875 }

```

14.2 Transpose

TODO: describe

TODO: describe

```

3876 \let\eql@transpose@active\eql@false
3877 \def\eql@transpose@end{\eql@transpose@end}

```



```

3878 \def\eql@transpose@skip{&\eqnpunct{}}
3879 \def\eql@transpose@complete{%
3880   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

```

TODO: describe

```

3881 \def\eql@transpose{%
3882   \eql@totalcolumns@\z@
3883   \eql@totalrows@\z@
3884   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
3885   \eql@scan@reg@{ }%
3886   \eql@row@\z@
3887   \eql@transpose@output@row
3888 }

```

TODO: describe

```

3889 \def\eql@transpose@save@col#1{%
3890   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
3891     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}

```

TODO: describe

```

3892 \def\eql@transpose@scan@col#1\&{%
3893   \def\eql@tmpa{#1}%
3894   \ifx\eql@tmpa\eql@transpose@end\else
3895     \advance\eql@totalcolumns@\@ne
3896     \eql@row@\z@
3897     \let\eql@transpose@data@col\@empty
3898     \eql@transpose@scan@row#1\\eql@transpose@end\\
3899     \ifnum\eql@row@>\eql@totalrows@
3900       \eql@totalrows@\eql@row@
3901     \fi
3902     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
3903     \expandafter\eql@transpose@scan@col
3904   \fi
3905 }

```

TODO: describe

```

3906 \def\eql@transpose@append@row#1{%
3907   \advance\eql@row@\@ne
3908   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}

```

TODO: describe

```

3909 \def\eql@transpose@scan@row#1\\{%
3910   \def\eql@tmpa{#1}%
3911   \ifx\eql@tmpa\eql@transpose@end\else
3912     \ifx\eql@transpose@active+
3913       \eql@transpose@scan@cell#1&\eql@transpose@end&%
3914     \else
3915       \eql@transpose@append@row{#1}%
3916     \fi
3917     \expandafter\eql@transpose@scan@row
3918   \fi
3919 }

```

TODO: describe

```

3920 \def\eql@transpose@scan@cell#1&#2&{%
3921   \def\eql@tmpa{#2}%

```

```

3922 \ifx\eql@tmpa\eql@transpose@end
3923   \eql@transpose@append@row{#1}%
3924 \else
3925   \eql@transpose@append@row{#1&#2}%
3926   \expandafter\eql@transpose@scan@cell@next
3927 \fi
3928 }

```

TODO: describe

```

3929 \def\eql@transpose@scan@cell@next#1&{%
3930   \def\eql@tmpa{#1}%
3931   \ifx\eql@tmpa\eql@transpose@end\else
3932     \eql@transpose@append@row{&#1}%
3933     \expandafter\eql@transpose@scan@cell@next
3934   \fi
3935 }

```

TODO: describe

```

3936 \def\eql@transpose@output@row{%
3937   \ifnum\eql@row@<\eql@totalrows@
3938     \advance\eql@row@\@ne
3939     \eql@column@\z@
3940     \eql@transpose@output@col
3941     \ifnum\eql@row@<\eql@totalrows@
3942       \eql@scan@addto\%
3943     \fi
3944     \expandafter\eql@transpose@output@row
3945   \fi
3946 }

```

TODO: describe

```

3947 \def\eql@transpose@output@col{%
3948   \ifnum\eql@column@<\eql@totalcolumns@
3949     \advance\eql@column@\@ne
3950     \csname eql@transpose@data@col@\the\eql@column@\endcsname
3951     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
3952     \ifnum\eql@column@<\eql@totalcolumns@
3953       \eql@scan@addto{\eql@transpose@complete}%
3954     \fi
3955     \expandafter\eql@transpose@output@col
3956   \fi
3957 }

```

14.3 Measure

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

s@measure@line@begin

```

3958 \def\eql@columns@measure@line@begin{%
3959 (dev)\eql@dev{starting line \the\eql@row}%
3960   \global\eql@column@\z@
3961   \global\eql@line@height@\z@
3962   \global\eql@line@depth@\z@
3963   \eql@numbering@measure@line@begin
3964   \eql@hook@linein
3965 }

```

```

3966 \def\eql@columns@measure@cell{%
3967   \eql@cellwidth@\wd\eql@cellbox@
3968   \ifdefined\eql@frame@cmd
3969     \eql@frame@measure
3970     \advance\eql@cellwidth@\eql@frame@margin@
3971   \fi
3972   \ifdim\ht\eql@cellbox@>\eql@line@height@
3973     \global\eql@line@height@\ht\eql@cellbox@
3974   \fi
3975   \ifdim\dp\eql@cellbox@>\eql@line@depth@
3976     \global\eql@line@depth@\dp\eql@cellbox@
3977   \fi
3978   \ifnum\eql@column@=\@ne
3979     \eql@dimensions@startrow
3980   \fi
3981   \ifodd\eql@column@
3982     \eql@shape@pos@\tw@
3983   \else
3984     \eql@shape@pos@\z@
3985   \fi
3986   \eql@shape@amount@\z@
3987   \eql@dimensions@savecell
3988   \ifodd\eql@column@\else
3989     \eql@dimensions@savesep
3990   \fi
3991   \kern\eql@cellwidth@
3992 }

```

mns@measure@line@end

```

3993 \def\eql@columns@measure@line@end{%
3994   \eql@punct@apply@line
3995   \eql@hook@lineout
3996   &\omit
3997   \ifnum\eql@column@>\eql@totalcolumns@
3998     \global\eql@totalcolumns@\eql@column@
3999   \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

```

4000   \ifdefined\eql@frame@cmd
4001     \advance\eql@column@\@ne
4002     \wd\eql@cellbox@\z@
4003     \eql@columns@measure@cell
4004   \fi
4005   \eql@measure@tag
4006   \eql@measure@endrow
4007 }

```

\eql@columns@measure

```

4008 \def\eql@columns@measure{%
4009 <dev>\eql@dev@enter\eql@columns@measure
4010   \eql@totalcolumns@\z@
4011   \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end
4012   \setbox\z@\vbox{\measuring@true\halign{%

```

```

4013 &%
4014 \global\advance\eql@column@\@ne
4015 \global\let\eql@cell@container\@empty
4016 \global\setbox\eql@cellbox@\hbox{%
4017 \eql@strut@cell
4018 \@lign
4019 $\m@th\displaystyle
4020 \eql@hook@colin
4021 ##%
4022 \eql@class@innerleft
4023 \eql@hook@innerleft
4024 $%
4025 }%
4026 \eql@cell@container
4027 \hfil
4028 \eql@columns@measure@cell
4029 \global\let\eql@frame@prevcmd\eql@frame@cmd
4030 &%
4031 \eql@prevwidth@\wd\eql@cellbox@
4032 \let\eql@frame@cmd\eql@frame@prevcmd
4033 \global\advance\eql@column@\@ne
4034 \global\let\eql@cell@container\@empty
4035 \setbox\eql@cellbox@\hbox{%
4036 \eql@strut@cell
4037 \@lign
4038 $\m@th\displaystyle
4039 \eql@hook@innerright
4040 \eql@class@innerright@sel
4041 ##%
4042 \eql@punct@apply@col
4043 \eql@hook@colout
4044 $%
4045 }%
4046 \eql@cell@container
4047 \eql@columns@measure@cell
4048 \hfil
4049 \crrc

4050 \noalign{%
4051 \eql@hook@blockbefore
4052 }%
4053 \eql@hook@blockin
4054 \eql@scan@body

4055 \ifvmode\else
4056 \eql@punct@apply@block
4057 \eql@hook@blockout
4058 \eql@display@endline
4059 \cr
4060 \fi
4061 \noalign{%
4062 \eql@hook@blockafter
4063 }%

```

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

```

4064 \omit
4065 \eql@column@\@ne
4066 \eql@columns@completerow

```

```

4067   \cr
4068 }%

4069 \eql@measure@close

4070 \setbox\z@\vbox{%
4071   \unvbox\z@
4072   \unpenalty
4073   \global\setbox\@ne\lastbox
4074 }%
4075 \eql@totalwidth@\wd\@ne

TODO: why not recycle box contents altogether?!

4076 \let\eql@colwidth@tab\@empty
4077 \loop
4078   \setbox\@ne\hbox{%
4079     \unhbox\@ne
4080     \unskip
4081     \global\setbox\thr@\@lastbox
4082   }%
4083 \ifhbox\thr@
4084   \eql@colwidth@save{\wd\thr@}%
4085 \repeat

4086 (dev)\eql@dev@leave\eql@columns@measure
4087 }

```

14.4 Columns Placement

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

```

4088 \def\eql@columns@adjust{%
4089   \ifodd\eql@totalcolumns@
4090     \advance\eql@totalcolumns@\@ne
4091   \fi
4092   \eql@tagpos@adjust@eval
4093   \eql@adjust@calc@columns
4094 }

```

14.5 Print

TODO: describe

ms@print@line@begin

```

4095 \def\eql@columns@print@line@begin{%
4096 (dev)\eql@dev{starting line \the\eql@row@}%
4097   \global\eql@column@\z@
4098   \global\eql@line@pos@\eql@marginleft@
4099   \global\eql@line@width@\z@
4100   \global\eql@line@avail@\eql@totalwidth@
4101   \global\eql@line@height@\z@
4102   \global\eql@line@depth@\z@
4103   \eql@numbering@print@line@begin
4104   \eql@hook@linein
4105 }

```

l@columns@print@cell

```
4106 \def\eq@columns@print@cell{%
4107   \eq@cellwidth@\wd\eq@cellbox@
4108   \ifodd\eq@column@
4109     \ifdefined\eq@frame@cmd
4110       \eq@frame@measure
4111       \advance\eq@cellwidth@\eq@frame@margin@
4112     \fi
4113     \dimen@z@
4114   \else
4115     \advance\eq@cellwidth@-\eq@prevwidth@
```

draw a frame

```
4116   \ifdefined\eq@frame@cmd
4117     \eq@frame@measure
4118     \advance\eq@cellwidth@\eq@frame@margin@
4119     \advance\eq@prevwidth@\eq@frame@margin@
4120     \eq@frame@print
4121   \fi
```

update height and depth

```
4122   \ifdim\ht\eq@cellbox@>\eq@line@height@
4123     \global\eq@line@height@\ht\eq@cellbox@
4124   \fi
4125   \ifdim\dp\eq@cellbox@>\eq@line@depth@
4126     \global\eq@line@depth@\dp\eq@cellbox@
4127   \fi
```

print box

```
4128   \kern-\eq@prevwidth@
4129   \unhbox\eq@cellbox@
4130   \dimen@-\eq@cellwidth@
4131   \fi
```

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

```
4132   \advance\dimen@\eq@colwidth@get\eq@column@\relax
4133   \kern\dimen@
```

update available and used space

```
4134   \dimen@\eq@colwidth@get\eq@column@\relax
4135   \ifdim\eq@cellwidth@>z@
4136     \ifdim\eq@line@width@=z@
4137       \eq@line@avail@\eq@line@pos@
4138       \ifodd\eq@column@
4139         \advance\eq@line@avail@\dimen@
4140         \advance\eq@line@avail@-\eq@cellwidth@
4141       \fi
4142       \global\eq@line@avail@\eq@line@avail@
4143     \fi
4144     \eq@line@width@\eq@line@pos@
4145     \ifodd\eq@column@
4146       \advance\eq@line@width@\dimen@
4147     \else
4148       \advance\eq@line@width@\eq@cellwidth@
4149     \fi
```

```

4150   \global\eq@line@width@\eq@line@width@
4151   \fi
4152   \advance\eq@line@pos@\dimen@
4153   \ifodd\eq@column@\else
4154     \advance\eq@line@pos@\eq@colsep@
4155   \fi
4156   \global\eq@line@pos@\eq@line@pos@
4157 }

```

```

4158 \def\eq@columns@print@trailright{%
4159   &\omit
4160   \eq@prevwidth@\wd\eq@cellbox@
4161   \let\eq@frame@cmd\eq@frame@prevcmd
4162   \global\advance\eq@column@\@ne
4163   \eq@columns@print@cell
4164 }

```

lums@print@line@end

```

4165 \def\eq@columns@print@line@end{%
4166   \eq@punct@apply@line
4167   \eq@hook@lineout
4168 % \TODO add an even column with empty stuff if box processing deferred
4169   \ifodd\eq@column@
4170     \expandafter\eq@columns@print@trailright
4171   \fi
4172   \eq@columns@completerow
4173   \eq@columns@print@tag
4174 }

```

ql@columns@print@tag

```

4175 \def\eq@columns@print@tag{%
4176   \kern-\dimexpr\eq@totalwidth+\eq@colsep@\relax

```

determine first line available space

```

4177   \eq@display@firstavail@set\eq@line@avail@
4178   \eq@columns@overfull
4179   \eq@numbering@print@line@eval
4180   \if@eqnsw
4181     \eq@tagbox@make\eq@composetag@print
4182   \fi
4183   \eq@tagpos@print@line@eval
4184   \eq@tagbox@print@cell
4185   \eq@tagpos@print@line@end
4186 }

```

\eq@columns@print

```

4187 \def\eq@columns@print{%
4188 (dev)\eq@dev@enter\eq@columns@print
4189   \eq@shape@align@disable
4190   \eq@display@halign@init\eq@columns@print@line@begin
4191   \eq@display@halign@letcr\eq@columns@print@line@end
4192   \tabskip\eq@marginleft@

4193   \halign{%
4194     &%
4195     \global\advance\eq@column@\@ne

```

```

4196 \global\let\eqL@cell@container\@empty
4197 \global\setbox\eqL@cellbox@\hbox{%
4198 \eqL@strut@cell
4199 \@lign
4200 $\m@th\displaystyle
4201 \eqL@hook@colin
4202 ##%
4203 \eqL@class@innerleft
4204 \eqL@hook@innerleft
4205 \eqL@tagging@mathsave
4206 $%
4207 \eqL@tagging@mathaddlast
4208 }%
4209 \eqL@cell@container
4210 \hfil
4211 \eqL@columns@print@cell
4212 \global\let\eqL@frame@prevcmd\eqL@frame@cmd
4213 \tabskip\z@skip
4214 &%
4215 \eqL@prevwidth@\wd\eqL@cellbox@
4216 \let\eqL@frame@cmd\eqL@frame@prevcmd
4217 \global\advance\eqL@column@\@ne
4218 \global\let\eqL@cell@container\@empty
4219 \setbox\eqL@cellbox@\hbox{%
4220 \unhbox\eqL@cellbox@
4221 \eqL@strut@cell
4222 \@lign
4223 $\m@th\displaystyle
4224 \eqL@hook@innerright
4225 \eqL@class@innerright@sel
4226 ##%
4227 \eqL@punct@apply@col
4228 \eqL@hook@colout
4229 \eqL@tagging@mathsave
4230 $%
4231 \eqL@tagging@mathaddlast
4232 }%
4233 \eqL@cell@container
4234 \eqL@columns@print@cell
4235 \hfil
4236 \tabskip\eqL@colsep@\relax
4237 \crr

4238 \noalign{%
4239 \eqL@display@halign@start
4240 \eqL@numbering@print@block@begin
4241 \eqL@hook@blockbefore
4242 }%
4243 \eqL@hook@blockin
4244 \eqL@scan@body
4245 \ifvmode\else
4246 \relax
4247 \eqL@punct@apply@block
4248 \eqL@hook@blockout
4249 \eqL@display@endline
4250 \cr
4251 \fi
4252 \noalign{%
4253 \eqL@hook@blockafter

```



```

4254     \eql@display@halign@end
4255 (dev)\eql@dev@leave\eql@columns@print
4256     }%
4257     \eql@tagging@tablesavealign
4258     }%
4259 }

```

15 Interface

15.1 Scanning the Equation Body

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

Token Register.

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

```
4260 \newtoks\eql@scan@reg@
```

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

```

4261 \def\eql@scan@body@dump{\the\eql@scan@reg@}
4262 \def\eql@scan@body@rescan{%
4263   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
4264 \let\eql@scan@body\eql@scan@body@dump

```

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

```
4265 \long\def\eql@scan@addto#1{\eql@scan@reg@\expandafter{\the\eql@scan@reg@#1}}
```

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

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

```

4266 \def\eql@scan@env#1{%
4267 (dev)\eql@dev@enter\eql@scan@env
4268   \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenenv}}%
4269   \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`:

```

4270 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
4271 \expandafter\let\expandafter\eql@scan@org\eql@scan@iterate
4272 \ifdefined\eql@scan@par
4273   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
4274 \else
4275   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
4276 \fi
4277 \eql@scan@iterate
4278 }

```

`\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:

```

4279 \long\def\eql@scan@env@iterate#1\end#2{%
4280   \edef\eql@scan@stack{%
4281     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4282   \ifx\@empty\eql@scan@stack
4283     \@checkend{#2}%
4284     \eql@scan@addto{#1}%
4285     \expandafter\let\eql@scan@iterate\eql@scan@env@org
4286 (dev)\eql@dev@leave\eql@scan@env
4287   \expandafter\eql@scan@end
4288   \else
4289     \eql@scan@addto{#1\end{#2}}%
4290     \expandafter\eql@scan@iterate
4291   \fi
4292 }

```

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

```

4293 \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.

```

4294 \long\def\eql@scan@env@count#1\begin#2{%
4295   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4296 }

```

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

```

4297 \def\eql@scan@env@cancel{%
4298   \@namedef{end\@currenvir}{\ignorespacesafterend}%
4299 }

```

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 `\]`:

```

4300 \def\eql@scan@sqr#1{%
4301 (dev)\eql@dev@enter\eql@scan@sqr

```

```

4302 \def\eql@scan@end{#1\}}%
4303 \eql@scan@reg@{\def\eql@scan@stack{b}%
4304 \let\eql@scan@sqr@org\[%\}
4305 \ifdefined\eql@scan@par
4306   \let\[\eql@scan@sqr@iterate%\}
4307 \else
4308   \let\[\eql@scan@sqr@iterate@nopar%\}
4309 \fi
4310 \[%\}
4311 }

```

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

```

4312 \long\def\eql@scan@sqr@iterate#1\{%
4313 \edef\eql@scan@stack{%
4314   \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4315 \ifx\@empty\eql@scan@stack
4316   \let\[\eql@scan@sqr@org%\}
4317   \eql@scan@addto{#1}%
4318 (dev)\eql@dev@leave\eql@scan@sqr
4319   \expandafter\eql@scan@end
4320 \else
4321   \eql@scan@addto{#1\}}%
4322   \expandafter\[%\}
4323 \fi
4324 }

```

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

```

4325 \def\eql@scan@sqr@iterate@nopar#1\{\eql@scan@sqr@iterate#1\}

```

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

```

4326 \long\def\eql@scan@sqr@count#1\[#2{\%
4327 \ifx\#2\else b\expandafter\eql@scan@sqr@count\fi
4328 }

```

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

```

4329 \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 `\>`:

```

4330 \def\eql@scan@ang#1{%
4331 (dev)\eql@dev@enter\eql@scan@ang
4332 \def\eql@scan@end{#1\>}%
4333 \eql@scan@reg@{\def\eql@scan@stack{b}%
4334 \let\eql@scan@ang@org\<%\>
4335 \ifdefined\eql@scan@par
4336   \let\<\eql@scan@ang@iterate%\>
4337 \else
4338   \let\<\eql@scan@ang@iterate@nopar%\>
4339 \fi
4340 \<%\>
4341 }

```

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

```
4342 \long\def\eql@scan@ang@iterate#1\>{%
4343   \edef\eql@scan@stack{%
4344     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4345   \ifx\@empty\eql@scan@stack
4346     \let\<\eql@scan@ang@org%\>
4347     \eql@scan@addto{#1}%
4348 (dev)\eql@dev@leave\eql@scan@ang
4349   \expandafter\eql@scan@end
4350   \else
4351     \eql@scan@addto{#1\>}%
4352     \expandafter\<%\>
4353   \fi
4354 }
```

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

```
4355 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}
```

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

```
4356 \long\def\eql@scan@ang@count#1\<#2{%\>
4357   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4358 }
```

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

```
4359 \def\eql@equations@testall{\eql@parseopt\eql@equations@parseopt}
4360 \def\eql@equations@parseopt{%
4361   \ifx\eql@parseopt@token*%
4362     \let\eql@parseopt@next\eql@parseopt@nonumber
4363   \fi
4364   \ifx\eql@parseopt@token!%
4365     \let\eql@parseopt@next\eql@parseopt@donumber
4366   \fi
4367   \ifx\eql@parseopt@token/%
4368     \let\eql@parseopt@next\eql@parseopt@transpose
4369   \fi
4370   \ifx\eql@parseopt@token[%]
4371     \let\eql@parseopt@next\eql@parseopt@opt
4372   \fi
4373   \ifx\eql@parseopt@token\eql@atxi
4374     \let\eql@parseopt@next\eql@parseopt@label
4375   \fi
4376   \ifx\eql@parseopt@token\eql@atxii
4377     \let\eql@parseopt@next\eql@parseopt@label
4378   \fi
4379   \ifx\eql@parseopt@token.%
4380     \let\eql@parseopt@next\eql@parseopt@punctdot
4381   \fi
4382   \ifx\eql@parseopt@token,%
4383     \let\eql@parseopt@next\eql@parseopt@punctcomma
```

```

4384 \fi
4385 \ifx\eql@parseopt@token~%
4386   \let\eql@parseopt@next\eql@parseopt@punctoff
4387 \fi
4388 \ifx\eql@parseopt@token'%
4389   \let\eql@parseopt@next\eql@parseopt@punctall
4390 \fi
4391 \ifx\eql@parseopt@token-%
4392   \let\eql@parseopt@next\eql@parseopt@single
4393 \fi
4394 \ifx\eql@parseopt@token=%
4395   \let\eql@parseopt@next\eql@parseopt@lines
4396 \fi
4397 \ifx\eql@parseopt@token|%
4398   \let\eql@parseopt@next\eql@parseopt@columns
4399 \fi
4400 \ifx\eql@parseopt@token\label
4401   \let\eql@parseopt@next\eql@parseopt@end
4402 \fi
4403 \ifx\eql@parseopt@token\begin
4404   \let\eql@parseopt@next\eql@parseopt@end
4405 \fi
4406 }

```

`\eql@equations@processopt` The macro `\eql@equations@processopt` processes the options received by `\eqnadopt`. 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:

```

4407 \def\eql@equations@processopt{%
4408   \let\eql@tags@container@block\eql@tags@container@clear
4409   \let\eql@tags@frame@cmd\@firstofone
4410   \let\eql@skip@force@above\@undefined
4411   \let\eql@skip@force@below\@undefined
4412   \let\eql@skip@force@leave\@undefined
4413   \let\eql@display@linewidth\@undefined
4414   \let\eql@display@marginleft\@undefined
4415   \let\eql@display@marginright\@undefined
4416   \eql@abovespace@\z@skip
4417   \eql@belowspace@\z@skip
4418   \eql@displaybreak@prepen@\@MM
4419   \eql@displaybreak@postpen@\@MM
4420   \eql@nextopt@process{equations}%
4421   \let\eql@punct@block\eql@punct@main
4422   \let\eql@punct@main\relax
4423   \eql@indent@\glueexpr\eql@indent@val\relax
4424   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4425 }

```

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:

```

4426 \def\eql@single@cr@error{%
4427   \eql@error{Cannot use '\string\` within display equation.

```

```

4428     Please switch to equations environment}%
4429 }

```

`\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:

```

4430 \def\eql@single@start{%
4431   \eql@display@enter
4432   \eql@tagging@start
4433   \eql@dollar@dollar@begin
4434   \eql@display@adjust
4435   \eql@numbering@init
4436   \eql@stack@save@equations
4437   \eql@numbering@single@init
4438   \ifdefined\eql@single@cr@mode
4439     \let\\\eql@single@cr@mode
4440   \fi
4441   \ifdefined\eql@single@native
4442     \let\eql@single@start@sel\eql@single@start@native
4443     \let\eql@single@end@sel\eql@single@end@native
4444   \else
4445     \let\eql@single@start@sel\eql@single@start@print
4446     \let\eql@single@end@sel\eql@single@end@print
4447   \fi
4448   \eql@single@start@sel
4449 }

```

`\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:

```

4450 \def\eql@single@end{%
4451   \eql@punct@apply@block
4452   \eql@hook@eqout
4453   \eql@single@end@sel
4454   \eql@stack@restore
4455   \eql@dollar@dollar@end
4456   \eql@tagging@end
4457   \eql@display@leave
4458 }

```

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

```

4459 \def\eql@single@main{%
4460   \expandafter\eql@single@start
4461   \eql@scan@body
4462   \eql@single@end
4463 }

```

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

```

4464 \def\eql@mode@single{%
4465   \ifdefined\eql@single@doscan
4466     \let\eql@equations@main\eql@single@main
4467     \let\eql@equations@end\@empty

```

```

4468 \else
4469   \let\eqlequations@main\@undefined
4470   \let\eqlequations@end\eqleq@single@end
4471 \fi
4472 }

```

15.4 Multi-Line Main

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

```
4473 \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

```

4474 \def\eql@multi@main{%
4475   \eql@display@enter
4476   \eql@tagging@start
4477   \eql@dollar@begin
4478   \eql@display@adjust
4479   \eql@numbering@init
4480   \eql@stack@save@equations
4481   \ifdefined\eql@transpose@active
4482     \ifdefined\eql@multi@mode@lines\else
4483       \eql@transpose
4484     \fi
4485   \fi
4486   \ifdefined\eql@numbering@subeq@use
4487     \eql@numbering@subeq@init
4488   \fi
4489   \eql@display@init
4490   \let\intertext\eql@intertext
4491   \let\endintertext\endeql@intertext
4492   \eql@shape@align@enable

```

Now measure the given multi-line equations body:

```

4493   \ifdefined\eql@multi@mode@lines
4494     \eql@lines@measure
4495   \else
4496     \ifdefined\eql@ampproof@active
4497       \eql@ampproof
4498     \fi
4499     \eql@columns@measure
4500   \fi

```

If only a single equation number is used for subequation numbering, revert to normal equation numbering. If only a single column is used in columns mode, may fallback to lines mode. Switching from columns to lines mode, the width can be incorrect, expect only minor discrepancies, but for accurateness, should call `\eql@lines@measure`:

```

4501   \ifdefined\eql@numbering@subeq@use
4502     \eql@numbering@subeq@test
4503   \fi
4504   \ifdefined\eql@multi@mode@lines\else
4505     \ifdefined\eql@multi@lines@fallback
4506       \ifnum\eql@totalcolumns@=\@ne

```

```

4507     \let\eql@multi@mode@lines\eql@true
4508     \ifx\eql@multi@linesfallback\z@\else
4509         \eql@lines@measure
4510     \fi
4511 \fi
4512 \fi
4513 \fi

```

Adjust the multi-line equations body:

```

4514 \ifdefined\eql@multi@mode@lines
4515     \eql@lines@adjust
4516 \else
4517     \eql@columns@adjust
4518 \fi

```

Now print the multi-line equations body:

```

4519 \eql@display@print
4520 \eql@numbering@print@init
4521 \ifdefined\eql@multi@mode@lines
4522     \eql@lines@print
4523 \else
4524     \eql@columns@print
4525 \fi
4526 \eql@display@close

```

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

```

4527 \ifdefined\eql@numbering@subeq@use
4528     \eql@numbering@subeq@close
4529 \fi
4530 \eql@stack@restore
4531 \eql@dollar@dollar@end
4532 \eql@tagging@end
4533 \eql@display@leave
4534 }

```

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

```

\eql@mode@lines
4535 \def\eql@mode@columns{%
4536     \let\eql@equations@main\eql@multi@main
4537     \let\eql@equations@end\@empty
4538     \let\eql@multi@mode@lines\eql@false
4539 }
4540 \def\eql@mode@lines{%
4541     \let\eql@equations@main\eql@multi@main
4542     \let\eql@equations@end\@empty
4543     \let\eql@multi@mode@lines\eql@true
4544 }

```

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 `\eql@equations@start`:

```

4545 \newenvironment{equations}{%
4546 (dev)\eql@dev@enterenv
4547 \ifmmode
4548   \eql@error@mathmode{\string\begin{\@currenvir}}%
4549   \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4550 \else
4551   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4552   \expandafter\eql@equations@start
4553 \fi
4554 }{%
4555 \eql@equations@end
4556 \ignorespacesafterend
4557 (dev)\eql@dev@leaveenv
4558 }
4559 \eql@markline@amsthm@register{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`:

```

4560 \def\eql@equations@start{%
4561 \eql@equations@processopt
4562 \ifdefined\eql@equations@main
4563   \expandafter\eql@scan@env\expandafter\eql@equations@main
4564 \else
4565   \expandafter\eql@single@start
4566 \fi
4567 }

```

Square Brackets.

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

```

4568 \newenvironment{equations@sqr}{}{}
4569 \eql@markline@amsthm@register{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`:

```

4570 \protected\def\eql@equations@sqr@open{%
4571 \ifmmode
4572   \eql@error@mathmode{\string\[...\string\]}%
4573   \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
4574 \else
4575 (dev)\eql@dev@enter{\string\[...\string\]}%
4576   \expandafter\eqnadopt\expandafter{\eql@equations@sqr@opt}%
4577   \begin{equations@sqr}%
4578   \let\]\eql@equations@sqr@close
4579   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4580   \expandafter\eql@equations@sqr@start
4581 \fi
4582 }

```

`@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`:

```
4583 \def\eql@equations@sqr@start{%
4584   \eql@equations@processopt
4585   \ifdefined\eql@equations@main
4586     \expandafter\eql@scan@sqr\expandafter\eql@equations@main
4587   \else
4588     \expandafter\eql@single@start
4589   \fi
4590 }
```

`@equations@sqr@close` Definition for ‘\’:

```
4591 \protected\def\eql@equations@sqr@close{%
4592   \eql@equations@end
4593   (dev)\eql@dev@leave{\[...\string\]}%
4594   \end{equations@sqr}%
4595   \ignorespaces
4596 }
```

TODO: describe

```
\eql@sqr@open
\eql@sqr@close
4597 \let\eql@sqr@open\eql@equations@sqr@open
4598 \protected\def\eql@sqr@close{%
4599   \eql@error{'\string\'} may only close '\string\[']%
4600 }
```

Angle Brackets.

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

```
4601 \newenvironment{equations@ang}{}{}
4602 \newenvironment{equationsbox@ang}{}{}
4603 \eql@markline@amsthm@register{equations@ang}
```

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

```
4604 \protected\def\eql@ang@open{%
4605   (dev)\eql@dev@enter{\<...\string\>}%
4606   \ifmmode
4607     \expandafter\eqnadopt\expandafter{\eql@box@ang@opt}%
4608     \begin{equationsbox@ang}%
4609     \let\>\eql@box@ang@close
4610     \expandafter\eql@ampprotect\expandafter\eql@box@testall
4611     \expandafter\eql@box@start
4612   \else
4613     \expandafter\eqnadopt\expandafter{\eql@equations@ang@opt}%
4614     \begin{equations@ang}%
4615     \let\>\eql@equations@ang@close
4616     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4617     \expandafter\eql@equations@ang@start
4618   \fi
4619 }
```

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

```

4620 \protected\def\eq@ang@close{%
4621   \eq@error{'\string\>' may only close '\string\<'}%\>
4622 }

```

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

```

4623 \def\eq@equations@ang@start{%
4624   \eq@equations@processopt
4625   \ifdefined\eq@equations@main
4626     \expandafter\eq@scan@ang\expandafter\eq@equations@main
4627   \else
4628     \expandafter\eq@single@start
4629   \fi
4630 }

```

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

```

4631 \def\eq@equations@ang@close{%
4632   \eq@equations@end
4633   \end{equations@ang}%
4634 (dev)\eq@dev@leave{\<...\string\>}%
4635   \ignorespaces
4636 }

```

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

```

4637 \def\eq@box@ang@close{%
4638   \eq@box@end
4639   \end{equationsbox@ang}%
4640 (dev)\eq@dev@leave{\<...\string\>}%
4641   \ignorespaces
4642 }

```

16 Options

16.1 Selection Tools

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

```

4643 \def\eq@decide@abovebelow#1#2#3#4#5{%
4644   \eq@decide@select{#1}{#2}{#3}{%
4645     {,abovebelow,both,tb}{#4#5},%
4646     {above,top,t}{#4},%
4647     {below,bottom,b}{#5}}

```

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

```

4648 \def\eq@decide@situation#1#2#3#4{%
4649   \eq@decide@select{#1}{#2}{#3}{%
4650     {{long}{\def#4{0}}},%
4651     {{short}{\def#4{1}}},%
4652     {{cont}{\def#4{2}}},%
4653     {{par}{\def#4{3}}},%
4654     {{top}{\def#4{4}}},%
4655     {{noskip}{\def#4{5}}},%
4656     {{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:

```
4657 \eql@define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{}%
4658   \eql@mode@stacked}
4659 \eql@define@key{equationsbox}{aligned,align,al,columns,col}[]{}%
4660   \eql@mode@aligned}
4661 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
4662 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
4663 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
4664 \eql@define@key{setup}{boxangopt}[]{}%
4665   \def\eql@box@ang@opt{columns,#1}}
```

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

```
4666 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@single}
4667 \eql@define@key{equations}{gathered,gather,ga,lines,ln}[]{}%
4668   \eql@mode@lines}
4669 \eql@define@key{equations}{aligned,align,al,columns,col}[]{}%
4670   \eql@mode@columns}
4671 \eql@define@key{equations,setup}{transpose}[true]{}%
4672   \eql@decide@select{#3}{#2}{#1}{}%
4673     {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
4674     {{noamp,plain,restricted}\let\eql@transpose@active\eql@true}},%
4675     {{\eql@decide@true,amp,cont}\let\eql@transpose@active=+}}}}
4676 \eql@define@key{equations}{native}[true]{}%
4677   \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
4678   \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
4679 \eql@define@key{setup}{native}[true]{}%
4680   \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
4681 \eql@define@key{setup}{scanequation}[true]{}%
4682   \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
4683 \eql@define@key{setup}{sqropt}[]{}%
4684   \def\eql@equations@sqr@opt{equation,#1}}
4685 \eql@define@key{setup}{angopt}[]{}%
4686   \def\eql@equations@ang@opt{columns,#1}}
```

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

```
4687 \def\eql@keycat{equations,equationsbox,setup}
4688 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
4689 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@select{#3}{#2}{#1}{}%
4690   {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
4691   {{cell}\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
4692   {{tag}\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
4693   {\eql@decide@true
4694     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
4695 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}
```

Settings concerning page breaks:

```
4696 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{}%
```

```

4697   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
4698   {{high,3}{\eql@displaybreak@pre3}},%
4699   {{med,medium,2}{\eql@displaybreak@pre2}},%
4700   {{low,1}{\eql@displaybreak@pre1}},%
4701   {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
4702   {{default,inherit,-1}{\eql@displaybreak@pre\m@ne}}}}
4703 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4704   {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
4705   {{high,3}{\eql@displaybreak@post3}},%
4706   {{med,medium,2}{\eql@displaybreak@post2}},%
4707   {{low,1}{\eql@displaybreak@post1}},%
4708   {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
4709   {{default,inherit,-1}{\eql@displaybreak@post\m@ne}}}}
4710 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4711   \eql@decide@select{#3}{#2}{#1}{%
4712     {{full,4}{\eql@displaybreak@inter4}},%
4713     {{high,3}{\eql@displaybreak@inter3}},%
4714     {{med,medium,2}{\eql@displaybreak@inter2}},%
4715     {{low,1}{\eql@displaybreak@inter1}},%
4716     {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
4717 \eql@define@key{equations}{prepenalty}{%
4718   \eql@displaybreak@prepen\numexpr#1\relax}
4719 \eql@define@key{equations}{postpenalty}{%
4720   \eql@displaybreak@postpen\numexpr#1\relax}
4721 \eql@define@key{equations,setup}{interpenalty}{%
4722   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

4723 \eql@define@key{control}{vspace}[]{\eql@vspace@add{#1}}
4724 \eql@define@key{control}{vspace*}[]{\eql@vspace@addfixedbefore{#1}}
4725 \eql@define@key{control}{vspace!}[]{\eql@vspace@addfixedafter{#1}}
4726 \eql@define@key{control}{break}[4]{\eql@displaybreak@level[#{#1}]}
4727 \eql@define@key{control}{penalty}[]{\eql@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4728 \eql@define@key\eql@keycat{displayheight}[strut]{%
4729   \eql@decide@select{#3}{#2}{#1}{%
4730     {\eql@decide@false{\let\eql@display@height\undefined}},%
4731     {{strut}{\def\eql@display@height{\ht\eql@strutbox@}}},%
4732     {\relax{\def\eql@display@height{#1}}}}
4733 \eql@define@key\eql@keycat{displaydepth}[strut]{%
4734   \eql@decide@select{#3}{#2}{#1}{%
4735     {\eql@decide@false{\let\eql@display@depth\undefined}},%
4736     {{strut}{\def\eql@display@depth{\dp\eql@strutbox@}}},%
4737     {\relax{\def\eql@display@depth{#1}}}}

```

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

```

4738 \eql@define@key{equations}{noskip}[]{%
4739   \eql@decide@abovebelow{#3}{#2}{#1}%
4740   {\def\eql@skip@force@above{5}}%
4741   {\def\eql@skip@force@below{5}}}
4742 \eql@define@key{equations}{short}[above]{%
4743   \eql@decide@abovebelow{#3}{#2}{#1}%
4744   {\def\eql@skip@force@above{1}}%
4745   {\def\eql@skip@force@below{1}}}
4746 \eql@define@key{equations}{long}[]{%

```

```

4747 \eql@decide@abovebelow{#3}{#2}{#1}%
4748   {\def\eql@skip@force@above{0}}%
4749   {\def\eql@skip@force@below{0}}%
4750 \eql@define@key{equations}{medskip}[]{%
4751   \eql@decide@abovebelow{#3}{#2}{#1}%
4752   {\def\eql@skip@force@above{6}}%
4753   {\def\eql@skip@force@below{6}}%
4754 \eql@define@key{equations}{par}[par]{%
4755   \eql@decide@select{#3}{#2}{#1}{%
4756     {{default,}{\let\eql@skip@force@leave\undefined}},%
4757     {{cont,hmode}{\let\eql@skip@force@leave\z@}},%
4758     {{par,vmode}{\let\eql@skip@force@leave@one
4759       \ifdefined\eql@skip@force@below\else
4760         \def\eql@skip@force@below{3}%
4761         \fi}},%
4762     {{top}{\let\eql@skip@force@leave\tw@
4763       \ifdefined\eql@skip@force@below\else
4764         \def\eql@skip@force@below{4
4765         \fi}}}}

```

Specify vertical spacing explicitly:

```

4766 \eql@define@key{equations}{skip}{%
4767   \def\eql@skip@force@above{7}%
4768   \def\eql@skip@custom@above{#1}%
4769   \let\eql@skip@force@below\eql@skip@force@above
4770   \let\eql@skip@custom@below\eql@skip@custom@above}
4771 \eql@define@key{equations}{aboveskip}{%
4772   \def\eql@skip@force@above{7}%
4773   \def\eql@skip@custom@above{#1}}
4774 \eql@define@key{equations}{belowskip}{%
4775   \def\eql@skip@force@below{7}%
4776   \def\eql@skip@custom@below{#1}}
4777 \eql@define@key{equations}{abovespace}{%
4778   \advance\eql@abovespace@glueexpr#1\relax}
4779 \eql@define@key{equations}{belowspace}{%
4780   \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4781 \eql@define@key{intertext}{skip}{%
4782   \def\eql@skip@force@above{7}%
4783   \def\eql@skip@custom@above{#1}%
4784   \let\eql@skip@force@below\eql@skip@force@above
4785   \let\eql@skip@custom@below\eql@skip@custom@above}
4786 \eql@define@key{intertext}{aboveskip}{%
4787   \def\eql@skip@force@below{7}%
4788   \def\eql@skip@custom@below{#1}}
4789 \eql@define@key{intertext}{belowskip}{%
4790   \def\eql@skip@force@above{7}%
4791   \def\eql@skip@custom@above{#1}}
4792 \eql@define@key{intertext}{noskip}[]{%
4793   \eql@decide@abovebelow{#3}{#2}{#1}%
4794   {\def\eql@skip@force@below{5}}%
4795   {\def\eql@skip@force@above{5}}}
4796 \eql@define@key{intertext}{short}[]{%
4797   \eql@decide@abovebelow{#3}{#2}{#1}%
4798   {\def\eql@skip@force@below{1}}%
4799   {\def\eql@skip@force@above{1}}}
4800 \eql@define@key{intertext}{long}[]{%

```

```

4801 \eql@decide@abovebelow{#3}{#2}{#1}%
4802   {\def\eql@skip@force@below{0}}%
4803   {\def\eql@skip@force@above{0}}%
4804 \eql@define@key{intertext}{medskip}[]{}%
4805 \eql@decide@abovebelow{#3}{#2}{#1}%
4806   {\def\eql@skip@force@below{6}}%
4807   {\def\eql@skip@force@above{6}}%

```

Configure general vertical spacing behaviour for various situations:

```

4808 \eql@define@key{setup}{skip, longskip}{%
4809   \abovedisplayskip\glueexpr#1\relax
4810   \belowdisplayskip\abovedisplayskip
4811   \def\eql@skip@long@above{#1}%
4812   \let\eql@skip@long@below\eql@skip@long@above}
4813 \eql@define@key{setup}{aboveskip, abovelongskip}{%
4814   \abovedisplayskip\glueexpr#1\relax
4815   \def\eql@skip@long@above{#1}}
4816 \eql@define@key{setup}{belowskip, belowlongskip}{%
4817   \belowdisplayskip\glueexpr#1\relax
4818   \def\eql@skip@long@below{#1}}
4819 \eql@define@key{setup}{aboveshortskip}{%
4820   \abovedisplayshortskip\glueexpr#1\relax
4821   \def\eql@skip@short@above{#1}}
4822 \eql@define@key{setup}{belowshortskip}{%
4823   \belowdisplayshortskip\glueexpr#1\relax
4824   \def\eql@skip@short@below{#1}}
4825 \eql@define@key{setup}{tagskip}{%
4826   \def\eql@skip@tag@above{#1}%
4827   \let\eql@skip@tag@below\eql@skip@tag@above}
4828 \eql@define@key{setup}{abovetagskip}{%
4829   \def\eql@skip@tag@above{#1}}
4830 \eql@define@key{setup}{belowtagskip}{%
4831   \def\eql@skip@tag@below{#1}}
4832 \eql@define@key{setup}{medskip}{%
4833   \def\eql@skip@med@above{#1}%
4834   \let\eql@skip@med@below\eql@skip@med@above}
4835 \eql@define@key{setup}{abovemedskip}{%
4836   \def\eql@skip@med@above{#1}}
4837 \eql@define@key{setup}{belowmedskip}{%
4838   \def\eql@skip@med@below{#1}}
4839 \eql@define@key{setup}{abovetopskip}{%
4840   \def\eql@skip@top@above{#1}}
4841 \eql@define@key{setup}{belowtopskip}{%
4842   \def\eql@skip@top@below{#1}}
4843 \eql@define@key{setup}{aboveparskip}{%
4844   \def\eql@skip@par@above{#1}}
4845 \eql@define@key{setup}{belowparskip}{%
4846   \def\eql@skip@par@below{#1}}
4847 \eql@define@key{setup}{abovecontskip}{%
4848   \eql@decide@select{#3}{#2}{#1}{%
4849     {\hide}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%
4850     {\relax{\def\eql@skip@cont@above{#1}}}}%
4851 \eql@define@key{setup}{belowcontskip}{%
4852   \def\eql@skip@cont@below{#1}}
4853 \eql@define@key{setup}{shortmode}{%
4854   \eql@decide@select{#3}{#2}{#1}{%
4855     {\off, never, no}{\def\eql@skip@mode@short{0}}},%
4856     {\above, neverbelow, notbelow, belowoff}{\def\eql@skip@mode@short{1}}},%

```

```

4857   {{belowone,belowsingle}{\def\eq@skip@mode@short{2}}},%
4858   {{belowall,always,on}{\def\eq@skip@mode@short{3}}}}
4859 \eq@define@key{setup}{abovecontmode}{%
4860   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
4861 \eq@define@key{setup}{belowcontmode}{%
4862   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
4863 \eq@define@key{setup}{aboveparmode}{%
4864   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@above}
4865 \eq@define@key{setup}{belowparmode}{%
4866   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@below}
4867 \eq@define@key{setup}{abovetopmode}{%
4868   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@above}
4869 \eq@define@key{setup}{belowtopmode}{%
4870   \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@below}

```

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

```

4871 \def\eq@keycat{equations,subequations}
4872 \eq@define@key\eq@keycat{label}{\eq@tags@addblock@label{#1}}
4873 \eq@define@key\eq@keycat{labelname}{\eq@tags@addblock@name{#1}}
4874 \eq@define@key\eq@keycat{tag}{\eq@tags@addblock@tag{#1}}
4875 \eq@define@key\eq@keycat{tag*}{%
4876   \eq@tags@addblock@tagform@off\eq@tags@addblock@tag{#1}}
4877 \eq@define@key\eq@keycat{taglabel}{\eq@tags@addblock@ref{#1}}

```

TODO: describe

```

4878 \eq@define@key{control}{label}{\eq@tags@add@label{#1}}
4879 \eq@define@key{control}{labelname}{\eq@tags@add@name{#1}}
4880 \eq@define@key{control}{tag}{\eq@tags@add@tag{#1}}
4881 \eq@define@key{control}{tag*}{\eq@tags@add@tagform@off\eq@tags@add@tag{#1}}
4882 \eq@define@key{control}{taglabel}{\eq@tags@add@ref{#1}}
4883 \eq@define@key{control}{shifftag}{\eq@tags@add@raiseshift{#1}}
4884 \eq@define@key{control}{smashtag}{\eq@tags@add@raisesmash{#1}}
4885 \eq@define@key{control}{pushtag}{[]\eq@tags@add@forceraise}

```

TODO: describe

```

4886 \eq@define@key{setup}{labelname}{\protected@edef\eq@tags@name@generic{#1}}
4887 \eq@define@key{setup}{autolabel}[true]{%
4888   \eq@decide@bool{#3}{#2}{#1}\eq@tags@autolabel}
4889 \eq@define@key{setup}{autotag}[true]{%
4890   \eq@decide@bool{#3}{#2}{#1}\eq@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

4891 \def\eq@keycat{equations,setup}
4892 \eq@define@key\eq@keycat{tagmargin}[auto]{%
4893   \eq@decide@select{#3}{#2}{#1}{%
4894     {{auto,\eq@decide@false}{\let\eq@tagmargin@val\undefined}},%
4895     {\relax{\def\eq@tagmargin@val{#1}}}}}
4896 \eq@define@key\eq@keycat{tagmargin*}{%
4897   \settowidth\dimen@{#1}\edef\eq@tagmargin@val{\the\dimen@}}
4898 \eq@define@key\eq@keycat{tagmarginratio}{%
4899   \eq@tagmargin@ratio\dimexpr#1pt\relax}
4900 \eq@define@key\eq@keycat{tagmarginthreshold}{%
4901   \def\eq@tagmargin@threshold{#1}}
4902 \eq@define@key\eq@keycat{mintagsep}{\def\eq@tagsepmin@val{#1}}
4903 \eq@define@key\eq@keycat{mintagwidth}{%

```



```

4904 \settothewidth\dimen@{#1}\edef\eql@tagsepmin@val{\the\dimen@}
4905 \eql@define@key\eql@keycat{mintagwidth*}{\settothewidth\eql@tagwidthmin@{#1}}
4906 \eql@define@key\eql@keycat{tagsnap}{%
4907 \eql@decide@select{#3}{#2}{#1}{%
4908   {\eql@decide@false{\let\eql@tagpos@snap\z@}},%
4909   {\relax{\def\eql@tagpos@snap{#1}}}}

```

Tag Layout. Configure methods to declare equation tag layout:

```

4910 \def\eql@keycat{equations,setup}
4911 \eql@define@key\eql@keycat{tagbox,taglayout}{%
4912 \eql@tags@taglayout@set{#1}}
4913 \eql@define@key\eql@keycat{tagbox*,taglayout*}{%
4914 \eql@tags@taglayout@set@direct{#1}}
4915 \eql@define@key\eql@keycat{tagform}{%
4916 \eql@tags@tagform@set{#1}}
4917 \eql@define@key\eql@keycat{tagform*}{%
4918 \eql@tags@tagform@set@direct{#1}}
4919 \eql@define@key\eql@keycat{subeqtemplate}{%
4920 \def\eql@subequations@template####1####2{#1}%
4921 \eql@append\eql@subequations@template{\theparentequation{equation}}}

4922 \eql@define@key{control}{tagbox,taglayout}{%
4923 \global\eql@append\eql@tags@container{\eql@tags@taglayout@set{#1}}}
4924 \eql@define@key{control}{tagbox*,taglayout*}{%
4925 \global\eql@append\eql@tags@container{\eql@tags@taglayout@set@direct{#1}}}
4926 \eql@define@key{control}{tagform}{%
4927 \global\eql@append\eql@tags@container{\eql@tags@tagform@set{#1}}}
4928 \eql@define@key{control}{tagform*}{{###1}}{%
4929 \global\eql@append\eql@tags@container{\eql@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

4930 \def\eql@keycat{equations,setup}
4931 \eql@define@key\eql@keycat{numberline,number,num,numline,n}[all]{%
4932 \eql@decide@select{#3}{#2}{#1}{%
4933   {\eql@decide@false,0,*}{\let\eql@numbering@active\eql@false}},%
4934   {\eql@decide@true,!}{\let\eql@numbering@active\eql@true}},%
4935   {none,n,-}{\let\eql@numbering@mode\eql@numbering@mode@multi
4936     \let\eql@numbering@active\eql@false}},%
4937   {single,1}{\let\eql@numbering@mode\eql@numbering@mode@single
4938     \let\eql@numbering@active\eql@true}},%
4939   {multi,@}{\let\eql@numbering@mode\eql@numbering@mode@multi
4940     \let\eql@numbering@active\eql@true}},%
4941   {\relax{\eql@numbering@set{#1}}}}
4942 \eql@define@key\eql@keycat{nonumber,nn,*}[]{%
4943 \let\eql@numbering@active\eql@false}
4944 \eql@define@key\eql@keycat{donumber,dn,!}[]{%
4945 \let\eql@numbering@active\eql@true}
4946 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
4947 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
4948 \eql@define@key\eql@keycat{tags,eqno}{%
4949 \eql@decide@select{#3}{#2}{#1}{%
4950   {right,r}{\let\eql@tagsleft\eql@false}},%
4951   {left,l}{\let\eql@tagsleft\eql@true}}}}
4952 \eql@define@key\eql@keycat{evadetag,avoidtag}[true]{%
4953 \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}
4954 \eql@define@key\eql@keycat{tagbetween}[true]{%

```

```
4955 \eql@decide@bool{#3}{#2}{#1}\eql@tagpos@doconvert}
```

TODO: describe

```
4956 \eql@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
4957 \eql@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
4958 \eql@define@key{control}{numberhere}[]{\eql@numberhere}
4959 \eql@define@key{control}{numbernext}[]{\eql@numbernext}
```

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

```
4960 \def\eql@keycat{equations,setup}
4961 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
4962   {{center,c}{\let\eql@layoutleft\eql@false}},%
4963   {{left,l}{\let\eql@layoutleft\eql@true}}}}
4964 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
4965 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
4966 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin{#1}}
4967 \eql@define@key\eql@keycat{leftmargin*}{%
4968   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin{\the\dimen@}}
4969 \eql@define@key\eql@keycat{minleftmargin}{%
4970   \def\eql@layoutleftmarginmin{#1}}
4971 \eql@define@key\eql@keycat{maxleftmargin}{%
4972   \eql@decide@select{#3}{#2}{#1}{%
4973     {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
4974     {\relax{\def\eql@layoutleftmarginmax{#1}}}}}}

4975 \def\eql@keycat{equations,equationsbox}
4976 \eql@define@key\eql@keycat{margin}{%
4977   \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
4978 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}
4979 \eql@define@key\eql@keycat{marginright}{\def\eql@display@marginright{#1}}
4980 \eql@define@key{equations}{linewidth,width}{\def\eql@display@linewidth{#1}}
```

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

```
4981 \def\eql@keycat{equations,setup}
4982 \eql@define@key\eql@keycat{alignshrink}{\eql@decide@select{#3}{#2}{#1}{%
4983   {{max,full,4}{\eql@alignbadness@inf@bad}},%
4984   {{high,3}{\eql@alignbadness@54\relax}},%
4985   {{med,medium,2}{\eql@alignbadness@18\relax}},%
4986   {{low,1}{\eql@alignbadness@6\relax}},%
4987   {{0,\eql@decide@false}{\eql@alignbadness@z@}}}}
4988 \eql@define@key\eql@keycat{tagshrink}{\eql@decide@select{#3}{#2}{#1}{%
4989   {{max,full,4}{\eql@tagbadness@inf@bad}},%
4990   {{high,3}{\eql@tagbadness@54\relax}},%
4991   {{med,medium,2}{\eql@tagbadness@18\relax}},%
4992   {{low,1}{\eql@tagbadness@6\relax}},%
4993   {{0,\eql@decide@false}{\eql@tagbadness@z@}}}}
4994 \eql@define@key\eql@keycat{alignbadness}{\eql@alignbadness@numexpr#1\relax}
4995 \eql@define@key\eql@keycat{tagbadness}{\eql@tagbadness@numexpr#1\relax}
4996 \eql@define@key\eql@keycat{mincolsep}{\eql@decide@select{#3}{#2}{#1}{%
4997   {{0,\eql@decide@false}{\def\eql@colsepmin{val{0pt}}},%
4998   {\relax{\def\eql@colsepmin{val{#1}}}}}}
4999 \eql@define@key\eql@keycat{maxcolsep}{\eql@decide@select{#3}{#2}{#1}{%
5000   {\eql@decide@false{\def\eql@colsepmax{val{.5\maxdimen}}},%
```

```

5001   {\relax{\def\eq@colsepmax@val{#1}}}}
5002 \eq@define@key\eq@keycat{fulllength}[true]{%
5003   \eq@decide@bool{#3}{#2}{#1}\eq@columns@fulllength}

5004 \eq@define@key\eq@keycat{linesep}{\eq@decide@select{#3}{#2}{#1}{%
5005   {0,\eq@decide@false}{\def\eq@break@line@sep{Opt}}}},%
5006   {\relax{\def\eq@break@line@sep{#1}}}}
5007 \eq@define@key\eq@keycat{linesep*}{\eq@decide@select{#3}{#2}{#1}{%
5008   {0,\eq@decide@false}{\def\eq@break@line@shortsep{Opt}}}},%
5009   {\relax{\def\eq@break@line@shortsep{#1}}}}
5010 \eq@define@key\eq@keycat{equationsbox,setup}{colsep}{\eq@decide@select{#3}{#2}{#1}{%
5011   {0,\eq@decide@false}{\def\eq@box@colsep{Opt}}}},%
5012   {\relax{\def\eq@break@col@sep{#1}}}}%
5013   \let\eq@box@colsep\eq@break@col@sep}
5014 \eq@define@key\eq@keycat{equations}{colsep}{\eq@decide@select{#3}{#2}{#1}{%
5015   {0,\eq@decide@false}{\def\eq@break@col@sep{Opt}}}},%
5016   {\relax{\def\eq@break@col@sep{#1}}}}%
5017   \let\eq@colsepmin@val\eq@box@colsep
5018   \let\eq@colsepmax@val\eq@box@colsep
5019   \let\eq@box@colsep\eq@break@col@sep}
5020 \eq@define@key\eq@keycat{colsep*}{\eq@decide@select{#3}{#2}{#1}{%
5021   {0,\eq@decide@false}{\def\eq@break@col@shortsep{Opt}}}},%
5022   {\relax{\def\eq@break@col@shortsep{#1}}}}

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5023 \def\eq@keycat{equations,equationsbox,setup}
5024 \eq@define@key\eq@keycat{shape}[default]{\eq@shape@set{#1}}
5025 \eq@define@key\eq@keycat{padding,pad}[indent]{%
5026   \eq@decide@select{#3}{#2}{#1}{%
5027     {\max}{\let\eq@paddingleft@val\undefined}},%
5028     {\indent}{\def\eq@paddingleft@val{\eq@indent@val}}},%
5029     {0,\eq@decide@false}{\def\eq@paddingleft@val{Opt}}}},%
5030   {\relax{\def\eq@paddingleft@val{#1}}}}%
5031   \let\eq@paddingright@val\eq@paddingleft@val}
5032 \eq@define@key\eq@keycat{padleft}[indent]{%
5033   \eq@decide@select{#3}{#2}{#1}{%
5034     {\max}{\let\eq@paddingleft@val\undefined}},%
5035     {\indent}{\def\eq@paddingleft@val{\eq@indent@val}}},%
5036     {0,\eq@decide@false}{\def\eq@paddingleft@val{Opt}}}},%
5037   {\relax{\def\eq@paddingleft@val{#1}}}}
5038 \eq@define@key\eq@keycat{padright}[indent]{%
5039   \eq@decide@select{#3}{#2}{#1}{%
5040     {\max}{\let\eq@paddingright@val\undefined}},%
5041     {\indent}{\def\eq@paddingright@val{\eq@indent@val}}},%
5042     {0,\eq@decide@false}{\def\eq@paddingright@val{Opt}}}},%
5043   {\relax{\def\eq@paddingright@val{#1}}}}
5044 \eq@define@key\eq@keycat{indent}[2em]{%
5045   \def\eq@indent@val{#1}}

```

TODO: describe

```

5046 \eq@define@key{control}{align}[]{%
5047   \eq@decide@select{#3}{#2}{#1}{%
5048     {l,left}{\global\eq@append\eq@cell@container{\eq@shape@pos@z}},%
5049     {c,center}{\global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}},%
5050     {r,right}{\global\eq@append\eq@cell@container{\eq@shape@pos@tw}}}}
5051 \eq@define@key{control}{shift,shifto}[]{%
5052   \eq@decide@select{#3}{#2}{#1}{%

```

```

5053    {\*,indent}\@shape@alignamount@set{\@indent}},%
5054    {\!,outdent}\@shape@alignamount@set{-\@indent}},%
5055    {\relax\@shape@alignamount@set{#1}}}}
5056 \@define@key{control}{shift*,shiftby}[]{\@shape@alignamount@add{#1}}

```

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

```

5057 \def\@keycat{equations,equationsbox,setup}
5058 \@define@key\@keycat{classout}{\@class@innerleft@set{#1}}
5059 \@define@key\@keycat{classin}{\@class@innerright@set{#1}}
5060 \@define@key\@keycat{classlead,classin*}{\@class@innerlead@set{#1}}
5061 \@define@key\@keycat{ampeq}[]{\@class@ampeq}
5062 \@define@key\@keycat{eqamp}[]{\@class@eqamp}
5063 \@define@key\@keycat{class}{\@decide@select{#3}{#2}{#1}{%
5064   {ampeq,amprel,eqafter,beforerel}\@class@ampeq},%
5065   {eqamp,relamp,eqbefore,afterrel}\@class@eqamp}}

```

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

```

5066 \def\@punct@all#1#2#3#4#5\@punct@end{%
5067   \def\@tmp{#4}\def\@tmpa{1}%
5068   \ifx\@tmp\@punct
5069     \ifnum#5=1111\relax
5070       \@punct@set\@punct@col{#1}%
5071       \@punct@set\@punct@line{#2}%
5072       \@punct@set\@punct@main{#3}%
5073     \else\ifnum#5=111\relax
5074       \@punct@set\@punct@line{#1}%
5075       \@punct@set\@punct@main{#2}%
5076     \else\ifnum#5=11\relax
5077       \@punct@set\@punct@main{#1}%
5078     \else
5079       \let\@punct@col\@empty
5080       \let\@punct@line\@empty
5081       \let\@punct@main\@empty
5082     \fi\fi\fi
5083   \else
5084     \@error{Too many arguments to punctall}%
5085   \fi
5086 }

```

TODO: describe

```

5087 \def\@keycat{equations,equationsbox,setup}
5088 \@define@key\@keycat{punctsep}[\,]{\def\@punct@sep{#1}}
5089 \@define@key\@keycat{punct}[\.]{\@punct@set\@punct@main{#1}}
5090 \@define@key\@keycat{punct*}[]{\let\@punct@main\relax}
5091 \@define@key\@keycat{punctline}[\,]{\@punct@set\@punct@line{#1}}
5092 \@define@key\@keycat{punctline*}[]{\let\@punct@line\relax}
5093 \@define@key\@keycat{punctcol}[\,]{\@punct@set\@punct@col{#1}}
5094 \@define@key\@keycat{punctcol*}[]{\let\@punct@col\relax}
5095 \@define@key\@keycat{punctall}[\,]{\@punct@all#111111\@punct@end}

5096 \@define@key{control}{punctsep}[\,]{\def\@punct@sep{#1}}
5097 \@define@key{control}{punct}[\.]{\@punct@set\@punct@block{#1}%
5098   \@punct@set\@punct@line{#1}\@punct@set\@punct@col{#1}}
5099 \@define@key{control}{punct*}[]{\let\@punct@block\relax}
5100 \@define@key{control}{punctapply}[]{\@punct@apply@block}

```

Frames. **TODO:** describe

```
5101 \eql@define@key{equationsbox}{frame}[\fbox]{%
5102   \def\eql@box@frame{#1}%
5103   \ifx\eql@box@frame\empty\let\eql@box@frame\@firstofone\fi}
5104 \eql@define@key{equationsbox}{wrap}[]{\eql@box@wrap#1}
```

TODO: describe

```
5105 \eql@define@key{control}{framecell}[\fbox]{%
5106   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}}}
5107 \eql@define@key{control}{frametag}[\fbox]{%
5108   \global\eql@append\eql@tags@container{\def\eql@tags@frame@cmd{#1}}}}
```

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

```
5109 \eql@define@key{equations,equationsbox}{alt}{}
```

Injections.

```
5110 \eql@define@key{control}{inject}{%
5111   \global\eql@append\eql@interline@container{%
5112     \eql@append\eql@display@injectbefore{#1}}}}
5113 \eql@define@key{control}{inject*}{%
5114   \global\eql@append\eql@interline@container{%
5115     \eql@append\eql@display@injectafter{#1}}}}
5116 \eql@define@key{control}{markline}[]{\eql@markline@inject{#1}}
5117 \eql@define@key{control}{markline*}[]{\eql@markline@inject{push,#1}}
5118 \eql@define@key{control}{qed}[]{\eql@markline@inject{qed,#1}}
5119 \eql@define@key{control}{qed*}[]{\eql@markline@inject{qed,push,#1}}
```

TODO: describe

```
5120 \eql@define@key{markline}{pos}{%
5121   \eql@decide@select{#3}{#2}{#1}{%
5122     {{below,push}{\let\eql@markline@pos\eql@markline@pos@below}},%
5123     {{baseline}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
5124     {{bottom}{\let\eql@markline@pos\eql@markline@pos@bottom}}}}
5125 \eql@define@key{markline}{below,push}[]{%
5126   \let\eql@markline@pos\eql@markline@pos@below}
5127 \eql@define@key{markline}{baseline}[]{%
5128   \let\eql@markline@pos\eql@markline@pos@baseline}
5129 \eql@define@key{markline}{bottom}[]{%
5130   \let\eql@markline@pos\eql@markline@pos@bottom}
5131 \eql@define@key{markline}{shift}{\def\eql@markline@shift{#1}}
5132 \eql@define@key{markline}{symbol}{\def\eql@markline@symbol{#1}}
5133 \eql@define@key{markline}{qed}[]{\let\eql@markline@symbol\eql@markline@qed}
5134 \eql@define@key{setup}{marksymbol}{\def\eql@markline@symbol{#1}}
5135 \eql@define@key{setup}{qedsymbol}{\def\eql@markline@qed{#1}}
5136 \eql@define@key{setup}{markpos}{%
5137   \eql@decide@select{#3}{#2}{#1}{%
5138     {{below}{\let\eql@markline@pos\eql@markline@pos@below}},%
5139     {{baseline}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
5140     {{bottom}{\let\eql@markline@pos\eql@markline@pos@bottom}}}}
```

Global Switches. Set global switches:

```
5141 \let\eql@multi@lines@fallback\eql@false
```

```

5142 \let\eql@scan@par\eql@false
5143 \let\eql@single@cr@mode\eql@false
5144 \let\eql@ampproof@active\eql@false

5145 \eql@define@key{equations,setup}{linesfallback}[true]{%
5146   \eql@decide@select{#3}{#2}{#1}{%
5147     {\eql@decide@false{\let\eql@multi@linesfallback\eql@false}},%
5148     {\reuse,lean}{\let\eql@multi@linesfallback\z@}},%
5149     {\measure,full,\eql@decide@true}{\let\eql@multi@linesfallback\eql@true}}}}
5150 \eql@define@key{setup}{ampproof}[true]{%
5151   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
5152 \eql@define@key{equations,setup}{equationcr}{%
5153   \eql@decide@select{#3}{#2}{#1}{%
5154     {\eql@decide@false{\let\eql@single@cr@mode\eql@false}},%
5155     {\eql@decide@true,break}{\let\eql@single@cr@mode\eql@break@line}},%
5156     {\error,verbose}{\let\eql@single@cr@mode\eql@single@cr@error}}}}
5157 \eql@define@key{setup}{modifierwarning}[true]{%
5158   \eql@decide@select{#3}{#2}{#1}{%
5159     {\eql@decide@false{\let\eql@parseopt@warn\@empty}},%
5160     {\eql@decide@true{\let\eql@parseopt@warn\eql@warn@parseopt}},%
5161     {\verbose,+}{\let\eql@parseopt@warn\eql@warn@parseopt@verbose}}}}
5162 \let\eql@parseopt@warn\eql@warn@parseopt
5163 \eql@define@key{equations,setup}{rescan}[true]{%
5164   \eql@decide@if{#3}{#2}{#1}{%
5165     {\let\eql@scan@body\eql@scan@body@rescan},%
5166     {\let\eql@scan@body\eql@scan@body@dump}}}
5167 \eql@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5168   \eql@decide@bool{#3}{#2}{#1}\eql@scan@par}
5169 \eql@define@key{setup}{defaults}{%
5170   \eql@decide@select{#3}{#2}{#1}{%
5171     {\classic}{\eql@defaults@classic}},%
5172     {\eqnlines}{\eql@defaults@eqnlines}}}}

```

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

```

5173 \let\eql@provide@opt@env\tw@
5174 \let\eql@provide@opt@amsmathends\eql@true
5175 \let\eql@provide@opt@backup\eql@false
5176 \let\eql@provide@opt@ang\eql@true
5177 \let\eql@provide@opt@eqref\eql@true

5178 \eql@define@key{setup}{amsmathends}[true]{%
5179   \eql@error@packageoption{#2}%
5180   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
5181 \eql@define@key{setup}{backup}[true]{%
5182   \eql@error@packageoption{#2}%
5183   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
5184 \eql@define@key{setup}{env}[equation]{%
5185   \eql@error@packageoption{#2}%
5186   \eql@decide@select{#3}{#2}{#1}{%
5187     {\none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5188     {\equation,latex}{\let\eql@provide@opt@env@ne}},%
5189     {\amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}}
5190 \eql@define@key{setup}{ang}[true]{%
5191   \eql@error@packageoption{#2}%
5192   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5193 \eql@define@key{setup}{eqref}[true]{%
5194   \eql@error@packageoption{#2}%
5195   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Shortcut Options. **TODO:** describe

```

5196 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5197 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5198 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5199 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5200 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5201 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5202 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5203 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5204 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
5205 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
5206 \def\eql@parseopt@punctoff#1{\eqnaddopt{punct={}}\eql@parseopt@peek}
5207 \def\eql@parseopt@punctall#1#2{\eqnaddopt{punctall={#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 \TeX , \LaTeX and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

5208 \def\eql@defaults@classic{%
5209   \eqnlineset{numberline=all}%
5210   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5211   \eqnlineset{maxcolsep=off}%
5212   \eqnlineset{spread={\jot}}%
5213   \eqnlineset{tagmargin}%
5214   \eqnlineset{tagmarginratio=1}%
5215   \eqnlineset{tagmarginthreshold=0.5}%
5216   \eqnlineset{leftmargin={\leftmargini}}%
5217   \eqnlineset{padding=max}%
5218   \eqnlineset{evadetag=off}%
5219   \eqnlineset{displayheight=off}%
5220   \eqnlineset{displaydepth=off}%
5221   \eqnlineset{shortmode=belowsingle}%
5222   \eqnlineset{abovecontmode=short}%
5223   \eqnlineset{belowcontmode=short}%
5224   \eqnlineset{aboveparmode=long}%
5225   \eqnlineset{belowparmode=long}%
5226   \eqnlineset{abovetopmode=long}%
5227   \eqnlineset{belowtopmode=long}%
5228   \eqnlineset{abovelongskip={\abovedisplayskip}}%
5229   \eqnlineset{belowlongskip={\belowdisplayskip}}%
5230   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5231   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5232   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5233   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5234   \eqnlineset{abovecontskip=0pt}%
5235   \eqnlineset{belowcontskip=0pt}%
5236   \eqnlineset{aboveparskip=0pt}%
5237   \eqnlineset{belowparskip=0pt}%
5238   \eqnlineset{abovetopskip=0pt}%
5239   \eqnlineset{belowtopskip=0pt}%
5240   \eqnlineset{abovetagskip=0pt}%
5241   \eqnlineset{belowtagskip=0pt}%

```

```

5242 \eqnlineset{equationcr=off}%
5243 \eqnlineset{linesfallback=false}%
5244 }

```

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):

```

5245 \def\eq1@defaults@eqnlines{%
5246   \eqnlineset{numberline=all}%
5247   \eqnlineset{mintagsep=.5em}%
5248   \eqnlineset{maxcolsep=2em}%
5249   \eqnlineset{spread={0.2\normalbaselineskip}}%
5250   \eqnlineset{tagmargin}%
5251   \eqnlineset{tagmarginratio=.334}%
5252   \eqnlineset{tagmarginthreshold=0.5}%
5253   \eqnlineset{leftmargin={\leftmargini}}%
5254   \eqnlineset{padding=0pt}%
5255   \eqnlineset{evadetag}%
5256   \eqnlineset{displayheight=strut}%
5257   \eqnlineset{displaydepth=strut}%
5258   \eqnlineset{shortmode=above}%
5259   \eqnlineset{abovecontmode=noskip}%
5260   \eqnlineset{belowcontmode=long}%
5261   \eqnlineset{aboveparmode=long}%
5262   \eqnlineset{belowparmode=long}%
5263   \eqnlineset{abovetopmode=noskip}%
5264   \eqnlineset{belowtopmode=long}%
5265   \eqnlineset{longskip={0.75\normalbaselineskip
5266     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5267   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5268     plus 0.25\normalbaselineskip}}%
5269   \eqnlineset{belowshortskip={0.0\normalbaselineskip
5270     plus 0.25\normalbaselineskip}}%
5271   \eqnlineset{medskip={0.4\normalbaselineskip
5272     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5273   \eqnlineset{abovecontskip=0pt}%
5274   \eqnlineset{belowcontskip=0pt}%
5275   \eqnlineset{aboveparskip=0pt}%
5276   \eqnlineset{belowparskip=0pt}%
5277   \eqnlineset{abovetopskip=0pt}%
5278   \eqnlineset{belowtopskip=0pt}%
5279   \eqnlineset{abovetagskip={0.25\normalbaselineskip
5280     minus 0.25\normalbaselineskip}}%
5281   \eqnlineset{belowtagskip={0.25\normalbaselineskip
5282     minus 0.25\normalbaselineskip}}%
5283   \eqnlineset{equationcr=break}%
5284   \eqnlineset{linesfallback=true}%
5285 }

```


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
\eql@provide@movestart
\provide@undefinecmd
\provide@undefineenv
5286 \def\eql@provide@movecmd#1#2{%
5287   \eql@letcs{#1\expandafter}\csname #2\endcsname
5288 }
5289 \def\eql@provide@moveenv#1#2{%
5290   \eql@provide@movecmd{#1}{#2}%
5291   \eql@markline@amsthm@register{#1}%
5292   \ifcsname end#2\endcsname
5293     \eql@provide@movecmd{end#1}{end#2}%
5294   \fi
5295 }
5296 \def\eql@provide@movestart#1#2{%
5297   \eql@provide@moveenv{#1}{#2}%
5298   \ifcsname #2*\endcsname
5299     \eql@provide@moveenv{#1*}{#2*}%
5300   \fi
5301 }
5302 \def\eql@provide@undefinecmd#1{%
5303   \eql@letcs{#1}\undefined
5304 }
5305 \def\eql@provide@undefineenv#1{%
5306   \eql@provide@undefinecmd{#1}%
5307   \eql@provide@undefinecmd{end#1}%
5308 }
```

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:

```
5309 \def\eql@amsmath@endfix#1#2{%
5310   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
5311   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
5312     \eql@provide@movecmd{end#1}{end#2}%
5313   \fi
5314 }
```

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

```
5315 \def\eql@amsmath@fixends{%
5316   \eql@amsmath@after{%
5317     \eql@amsmath@endfix{gather*}{gather}%
5318     \eql@amsmath@endfix{multline*}{multline}%
5319   }
```

```

5319 \eql@amsmath@endfix{align*}{align}%
5320 \eql@amsmath@endfix{flalign}{align}%
5321 \eql@amsmath@endfix{flalign*}{align}%
5322 \eql@amsmath@endfix{alignat}{align}%
5323 \eql@amsmath@endfix{alignat*}{align}%
5324 \eql@amsmath@endfix{xalignat}{align}%
5325 \eql@amsmath@endfix{xalignat*}{align}%
5326 \eql@amsmath@endfix{xxalignat}{align}%
5327 \eql@amsmath@endfix{gathered}{aligned}%
5328 \eql@amsmath@endfix{alignedat}{aligned}%
5329 }
5330 }

```

Backup amsmath Environments. We can backup all amsmath environments *env* to *amsenv* so that they can be used in parallel if needed.

`provide@backup@amsmath` Copy an amsmath environment *env* to *amsenv* whenever amsmath is loaded: **TODO:**
amsthm

```

5331 \def\eql@provide@backup@amsmath#1{%
5332 \eql@amsmath@after{%
5333 \eql@provide@moveenv{ams#1}{#1}%
5334 \AddToHook{package/amsthm/after}{\eql@provide@movecmd{ams#1@qed}{#1@qed}}%
5335 }%
5336 }

```

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

```

5337 \def\eql@provide@backup@eqref{%
5338 \eql@amsmath@after{%
5339 \eql@provide@movecmd{amseqref}{eqref}%
5340 }%
5341 }

```

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

```

5342 \def\eql@provide@backup@multlined{%
5343 \AddToHook{package/mathtools/after}{%
5344 \eql@provide@moveenv{amsmultlined}{multlined}%
5345 }%
5346 }

```

`provide@backup@equation` The \LaTeX environment `equation` is overwritten by several packages to implement their adjustments. Here we cater for adjustments through `amsmath`, `hyperref` and the PDF tagging mechanism. Copy `equation` and `equation*` whenever `amsmath` is loaded. Whenever `hyperref` is loaded, and `amsmath` is not yet present, backup the original \LaTeX and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original \LaTeX `equation`. The PDF tagging mechanism registers `equation` upon `\begin{document}`. We thus need to register all copies of `equation` on our own, so that they can be used with their new names:

```

5347 \def\eql@provide@backup@equation{%
5348 \eql@amsmath@after{%
5349 \eql@provide@moveenv{amsequation}{equation}%
5350 \eql@tagging@register@env{amsequation}%
5351 \eql@provide@moveenv{amsequation*}{equation*}%

```

```

5352 \eql@tagging@register@env{amsequation*}%
5353 \AddToHook{package/amsthm/after}{%
5354 \eql@provide@movecmd{amsequation*@qed}{equation*@qed}}%
5355 }%
5356 \AddToHook{package/hyperref/after}{%
5357 \ifpackageloaded{amsmath}{}%
5358 \let\latexequation\H@equation
5359 \let\endlatexequation\H@endequation
5360 \eql@tagging@register@env{latexequation}%
5361 \eql@provide@moveenv{hyperrefequation}{equation}%
5362 \eql@tagging@register@env{hyperrefequation}%
5363 \AddToHook{package/amsthm/after}{%
5364 \eql@provide@movecmd{latexequation@qed}{equation@qed}%
5365 \eql@provide@movecmd{hyperequation@qed}{equation@qed}
5366 }%
5367 }%
5368 }%
5369 \@ifpackageloaded{amsmath}{%\@ifpackageloaded{hyperref}{}%
5370 \eql@provide@moveenv{latexequation}{equation}%
5371 \eql@tagging@register@env{latexequation}%
5372 \AddToHook{package/amsthm/after}{%
5373 \eql@provide@movecmd{latexequation@qed}{equation@qed}}%
5374 }%
5375 }

```

@backup@displaymath **TODO:** describe

```

5376 \def\eql@provide@backup@displaymath{%
5377 \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5378 \AddToHook{package/amsthm/after}{%
5379 \eql@provide@movecmd{latexdisplaymath@qed}{displaymath@qed}}%
5380 }

```

@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:

```

5381 \def\eql@provide@backup@subequations{%
5382 \eql@amsmath@after{%
5383 \eql@provide@moveenv{amssubequations}{subequations}%
5384 }%
5385 \AddToHook{package/hyperref/after}{%
5386 \AddToHook{cmd/amssubequations/before}{%
5387 {%
5388 \stepcounter{equation}%
5389 \protected@edef\theHparentequation{\theHequation}%
5390 \addtocounter{equation}{-1}%
5391 }%
5392 \AddToHook{cmd/amssubequations/after}{%
5393 {%
5394 \def\theHequation{\theHparentequation\alph{equation}}%
5395 \ignorespaces
5396 }%
5397 }%
5398 }

```

\eql@provide@backup Backup all amsmath environments:

```

5399 \def\eql@provide@backup{%

```

```

5400 \eql@provide@backup@eqref
5401 \eql@provide@backup@equation
5402 \eql@provide@backup@displaymath
5403 \eql@provide@backup@amsmath{gather}%
5404 \eql@provide@backup@amsmath{gather*}%
5405 \eql@provide@backup@amsmath{multline}%
5406 \eql@provide@backup@amsmath{multline*}%
5407 \eql@provide@backup@amsmath{align}%
5408 \eql@provide@backup@amsmath{align*}%
5409 \eql@provide@backup@amsmath{flalign}%
5410 \eql@provide@backup@amsmath{flalign*}%
5411 \eql@provide@backup@amsmath{alignat}%
5412 \eql@provide@backup@amsmath{alignat*}%
5413 \eql@provide@backup@amsmath{xalignat}%
5414 \eql@provide@backup@amsmath{xalignat*}%
5415 \eql@provide@backup@amsmath{xxalignat}%
5416 \eql@provide@backup@amsmath{aligned}%
5417 \eql@provide@backup@amsmath{aligned*}%
5418 \eql@provide@backup@amsmath{alignedat}%
5419 \eql@provide@backup@amsmath{alignedat*}%
5420 \eql@provide@backup@amsmath{gathered}%
5421 \eql@provide@backup@amsmath{gathered*}%
5422 \eql@provide@backup@multlined
5423 \eql@provide@backup@subequations
5424 }

```

Replacement amsmath Environments. TODO: describe

```

5425 \def\eql@alignat@gobblecol#1{%
5426 \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.)
5427 \newenvironment{eql@gathered}
5428 {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5429 \newenvironment{eql@multlined}
5430 {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5431 \newenvironment{eql@aligned}
5432 {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
5433 \newenvironment{eql@alignedat}
5434 {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
5435 {\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.)
5436 \newenvironment{eql@equation}
5437 {\eqnaddopt{equation}\equations}{\endequations}
5438 \newenvironment{eql@displaymath}
5439 {\eqnaddopt{equation,number}\equations}{\endequations}
5440 \newenvironment{eql@gather}
5441 {\eqnaddopt{lines}\equations}{\endequations}
5442 \newenvironment{eql@multline}
5443 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5444 {\endequations}
5445 \newenvironment{eql@align}

```

```

5446 {\eqnaddopt{columns}\equations}{\endequations}
5447 \newenvironment{eql@flalign}
5448 {\eqnaddopt{fulllength}\eql@align}{\endequations}
5449 \newenvironment{eql@alignat}
5450 {\eqnaddopt{colsep=off}\eql@xalignat}{\endequations}
5451 \newenvironment{eql@xalignat}
5452 {\eql@alignat@gobblecol\eql@align}{\endequations}
5453 \newenvironment{eql@xxalignat}
5454 {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
5455 \newenvironment{eql@equation*}
5456 {\eqnaddopt{nonumber}\eql@equation}{\endequations}
5457 \newenvironment{eql@gather*}
5458 {\eqnaddopt{nonumber}\eql@gather}{\endequations}
5459 \newenvironment{eql@multline*}
5460 {\eqnaddopt{nonumber}\eql@multline}{\endequations}
5461 \newenvironment{eql@align*}
5462 {\eqnaddopt{nonumber}\eql@align}{\endequations}
5463 \newenvironment{eql@flalign*}
5464 {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
5465 \newenvironment{eql@alignat*}
5466 {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
5467 \newenvironment{eql@xalignat*}
5468 {\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):

```

5469 \def\eql@provide@onlyonce#1#2{%
5470   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
5471   \ifx\eql@tmp\eql@tmpa
5472     \let\eql@tmp\@empty
5473   \fi
5474   \ifx\eql@tmp\@empty
5475     \let\eql@tmp\@undefined
5476     \ifx\@nodocument\relax
5477       \def\eql@tmp{#1}%
5478     \fi
5479     \ifcsname eql@provided@#1\endcsname
5480       \def\eql@tmp{#1}%
5481     \fi
5482     \eql@letcs{eql@provided@#1}\eql@true
5483   \else
5484     \def\eql@tmpa{*}%
5485     \ifx\eql@tmp\eql@tmpa
5486       \def\eql@tmp{#1}%
5487     \fi
5488   \fi
5489 }

```

`\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:

```

5490 \def\eql@provide@eqref#1{%
5491   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
5492   \ifx\eql@tmp\eql@tmpa
5493     \let\eql@tmp\@empty
5494   \fi
5495   \ifx\eql@tmp\@empty
5496     \eql@provide@onlyonce{eqref}{}%
5497   \else
5498     \def\eql@tmpa{*}%
5499     \ifx\eql@tmp\eql@tmpa
5500       \def\eql@tmp{eqref}%
5501     \else
5502       \edef\eql@tmp{\expandafter\@gobble\string#1}%
5503     \fi
5504   \fi
5505   \ifdefined\eql@tmp
5506     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{\eql@eqref}%
5507   \else
5508     \eql@amsmath@after{%
5509       \eql@provide@movecmd{eqref}{\eql@eqref}%
5510     }%
5511     \AddToHook{package/mathtools/after}{%
5512       \eql@provide@movecmd{eqref}{\eql@eqref}%
5513     }%
5514     \eql@provide@movecmd{eqref}{\eql@eqref}%
5515     \eql@amsmath@undefine\eqref
5516   \fi
5517 }

```

`\eql@provide@amsmath` Provide one of the amsmath environments and its star variant. Copy into place, and copy again when amsmath or mathtools are loaded. Remove definition before amsmath is loaded in the future to avoid an error:

```

5518 \def\eql@provide@amsmath#1#2{%
5519   \eql@provide@onlyonce{#1}{#2}%
5520   \ifdefined\eql@tmp
5521     \expandafter\eql@provide@movestar\expandafter{\eql@tmp}{\eql@#1}%
5522   \else
5523     \eql@amsmath@after{%
5524       \eql@provide@movestar{#1}{\eql@#1}%
5525     }%
5526     \AddToHook{package/mathtools/after}{%
5527       \eql@provide@movestar{#1}{\eql@#1}%
5528     }%
5529     \eql@provide@movestar{#1}{\eql@#1}%
5530     \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
5531     \ifcsname eql@#1*\endcsname
5532       \eql@amsmath@before{\eql@provide@undefineenv{#1*}}%
5533     \fi
5534   \fi
5535 }

```

`\eql@provide@multlined` Provide mathtools environment `multlined`. Copy into place, and copy again when amsmath or mathtools are loaded. Remove definition before mathtools is loaded in the future to avoid an error:

```

5536 \def\eql@provide@multlined#1{%
5537   \eql@provide@onlyonce{multlined}{#1}%
5538   \ifdefined\eql@tmp
5539     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@multlined}%
5540   \else
5541     \AddToHook{package/mathtools/after}{%
5542       \eql@provide@moveenv{multlined}{eql@multlined}%
5543     }%
5544     \eql@provide@moveenv{multlined}{eql@multlined}%
5545     \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5546       \eql@provide@undefineenv{multlined}}}%
5547   \fi
5548 }

```

`\eql@provide@equation` Provide the environment `equation` and its star variant. 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:

```

5549 \def\eql@provide@equation#1{%
5550   \eql@provide@onlyonce{equation}{#1}%
5551   \ifdefined\eql@tmp
5552     \expandafter\eql@provide@movestart\expandafter{\eql@tmp}{eql@equation}%
5553   \else
5554     \eql@amsmath@after{%
5555       \eql@provide@movestart{equation}{eql@equation}%
5556     }%
5557     \AddToHook{package/hyperref/after}{%
5558       \@ifpackageloaded{amsmath}{}%
5559       \eql@provide@moveenv{equation}{eql@equation}%
5560     }%
5561     \eql@provide@movestart{equation}{eql@equation}%
5562     \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5563     \ifdefined\eql@tagging@on
5564       \AddToHook{begindocument/end}{%
5565         \eql@provide@movestart{equation}{eql@equation}%
5566       }%
5567     \fi
5568   \fi
5569   \fi
5570 }

```

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

```

5571 \def\eql@provide@displaymath#1{%
5572   \eql@provide@onlyonce{displaymath}{#1}%
5573   \ifdefined\eql@tmp
5574     \expandafter\eql@provide@moveenv\expandafter{\eql@tmp}{eql@displaymath}%
5575   \else
5576     \eql@provide@moveenv{displaymath}{eql@displaymath}%
5577     \ifdefined\eql@tagging@on
5578       \AddToHook{begindocument/end}{%
5579         \eql@provide@moveenv{displaymath}{eql@displaymath}%
5580       }%
5581     \fi
5582   \fi
5583 }

```

`\eql@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:

```

5584 \def\eql@provide@subequations#1{%
5585   \eql@provide@onlyonce{subequations}{#1}%
5586   \ifdefined\eql@tmp
5587     \expandafter\eql@provide@moveenv
5588     \expandafter{\eql@tmp}{eql@subequations}%
5589   \else
5590     \eql@amsmath@after{%
5591       \eql@provide@moveenv{subequations}{eql@subequations}%
5592     }%
5593     \AddToHook{package/hyperref/after}{%
5594       \AddToHook{cmd/subequations/before}[hyperref]{}%
5595       \AddToHook{cmd/subequations/after}[hyperref]{}%
5596       \RemoveFromHook{cmd/subequations/before}[hyperref]%
5597       \RemoveFromHook{cmd/subequations/after}[hyperref]%
5598       \AddToHook{begindocument/end}{%
5599         \eql@provide@moveenv{subequations}{eql@subequations}%
5600       }%
5601     }%
5602     \eql@provide@moveenv{subequations}{eql@subequations}%
5603     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
5604   \fi
5605 }

```

`\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:

```

5606 \def\eql@provide@sqr{%
5607   \let\[ \eql@sqr@open
5608   \let\] \eql@sqr@close
5609   \eql@amsmath@after{%
5610     \let\[ \eql@sqr@open
5611     \let\] \eql@sqr@close
5612   }%
5613   \ifdefined\eql@tagging@on
5614     \AddToHook{begindocument/end}{%
5615       \let\[ \eql@sqr@open
5616       \let\] \eql@sqr@close
5617     }%
5618   \fi
5619 }

```

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

```

5620 \def\eql@provide@ang{%
5621   \let\< \eql@ang@open
5622   \let\> \eql@ang@close
5623 }

```

Interface.

`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

5624 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5625 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5626 \eql@define@key{provide}{gather}[]{\eql@provide@amsmath{gather}{#1}}
5627 \eql@define@key{provide}{multline}[]{\eql@provide@amsmath{multline}{#1}}
5628 \eql@define@key{provide}{align}[]{\eql@provide@amsmath{align}{#1}}
5629 \eql@define@key{provide}{flalign}[]{\eql@provide@amsmath{flalign}{#1}}
5630 \eql@define@key{provide}{alignat}[]{\eql@provide@amsmath{alignat}{#1}}
5631 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsmath{xalignat}{#1}}
5632 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsmath{xxalignat}{#1}}
5633 \eql@define@key{provide}{aligned}[]{\eql@provide@amsmath{aligned}{#1}}
5634 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsmath{alignedat}{#1}}
5635 \eql@define@key{provide}{gathered}[]{\eql@provide@amsmath{gathered}{#1}}
5636 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5637 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5638 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
5639 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
5640 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
5641 \eql@define@key{provide}{tagform}[]{%
5642   \def\tagform@##1{\maketag@@{\eql@tags@tagform{#1}}}}
5643 \eql@define@key{provide}{maketag}[]{%
5644   \def\maketag@@##1{\eql@tags@taglayout{##1}}}
```

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

```

5645 \newcommand{\eqnlinesprovide}[1]{%
5646 (dev)\eql@dev@start\eqnlinesprovide
5647   \eql@setkeys{provide}{#1}%
5648   \ignorespaces
5649 }
```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
5650 \let\eql@error@packageoption@gobble
```

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

```

5651 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
5652 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
5653 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

5654 \eql@defaults@eqnlines
5655 \eql@mode@columns
5656 \eql@mode@aligned
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the T_EX conditional parsing mechanism:

```

5657 \ifdefined\tagsleft@true\else
5658   \expandafter\newif\csname iftagsleft@\endcsname
```

```

5659 \fi
5660 \ifdefined\@fleqntrue\else
5661   \expandafter\newif\csname if@fleqn\endcsname
5662 \fi

```

Import amsmath switches leqno as tagsleft and fleqn as left:

```

5663 \eql@amsmath@after{%
5664   \ifnum\eql@provide@opt@env=\tw@
5665     \iftagsleft@
5666       \eqnlineset{tags=left}%
5667     \else
5668       \eqnlineset{tags=right}%
5669     \fi
5670   \if@fleqn
5671     \eqnlineset{layout=left}%
5672   \else
5673     \eqnlineset{layout=center}%
5674   \fi
5675 \fi
5676 }

```

Process package options:

```
5677 \ProcessOptions
```

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

```

5678 \def\eql@error@packageoption#1{%
5679   \eql@error{may only use '#1' as a package option}%
5680 }

```

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

```
5681 \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:

```
5682 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

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

```

5683 \ifnum\eql@provide@opt@env>\z@
5684   \eqnlinesprovide{equation,sqr,displaymath}
5685 \fi

```

Provide amsmath equation environments if desired:

```

5686 \ifnum\eql@provide@opt@env=\tw@
5687   \eqnlinesprovide{%
5688     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
5689     multlined,gathered,aligned,alignedat,%
5690     subequations}
5691 \fi

```

Provide symbolic shortcut `\<...>` if desired:

```
5692 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
5693 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```